Automotive E/E and Automotive Software Technology

4

This chapter begins with an overview of mechatronic systems in the automotive domain in Sect. 4.1. Section 4.2 focuses on automotive electronics, taking into account body, chassis, comfort, driver assistance electronics, electronic control units (ECUs), and entertainment/infotainment electronics, as well as sensor technology. In Sect. 4.3, electrical and electronic (E/E) architectures and bus system requirements are introduced, with emphasis on disciplined approaches to their design. Section 4.4 discusses the concept of functional safety. Thereafter, Sect. 4.5 focuses on automotive software engineering, taking into account the increasing role of software content and product complexity, model-based software development, and hardware-in-the-loop (HIL) tests. Section 4.6 refers to the AUTomotive Open System Architecture (AUTOSAR) platform and Sect. 4.7 to the AUTOSAR Adaptive Platform. In Sect. 4.8, the GENIVI Alliance®, essential for telematic and infotainment components, is introduced. Section 4.9 provides examples of advanced driver assistance systems (ADASs) (see also Chap. 11), and Sect. 4.10 looks at future trends. Section 4.11 contains a comprehensive set of questions on automotive E/E and automotive software engineering, and finally followed by references and suggestions for further reading.

4.1 Mechatronic Systems in the Car

Mechatronics is an interdisciplinary engineering concept that synergistically combines mechanics, electronics, and software technologies in an integrated and optimized product design. The name itself was first used in Japan in the 1960s. Today, mechatronics is widely recognized as an engineering discipline with different application domains, as shown in Fig. 4.1. Mechatronics has huge potential in the automotive industry market as its products are upgraded and differentiated through innovation. For instance, mechatronic product features, such as power seats, electronic mirrors, automatic climate control, and others, facilitate memory functions

D. P. F. Möller, R. E. Haas, *Guide to Automotive Connectivity and Cybersecurity*,
Computer Communications and Networks,
https://doi.org/10.1007/978-3-319-73512-2_4

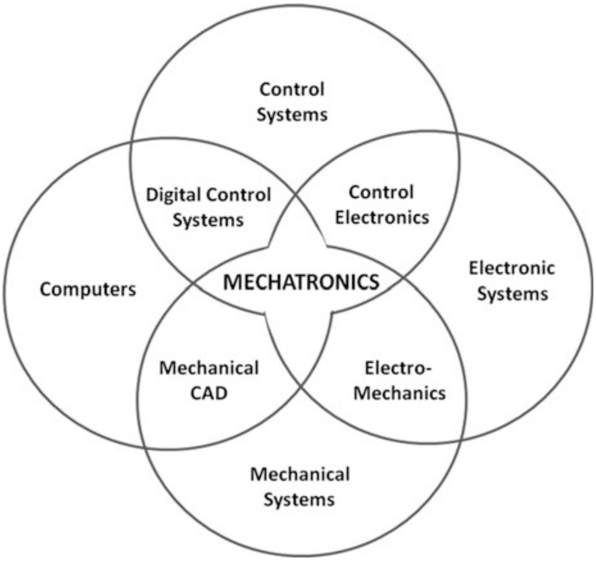


Fig. 4.1 Interdisciplinary disciplines of mechatronics

and automatic actuation in these products. Hence, mechatronics can be seen as the engineering answer to the manifold innovative demands of the automotive market which deal with the analysis, design, implementation, and test of electromechanical systems that are controlled by electronics and embedded software, such as antilock braking, power train control, door locking mechanism, suspension, and others.

As can be seen in Fig. 4.1, there are some fundamental trends that have significantly shaped the work of mechatronics engineers. These are primarily driven by advances in networking, modeling, simulation, and control, as well as embedded system design. The three major trends in mechatronics today are increasing software content, networked systems, and intelligent controls.

- *Rapid Increase of Software Content:* There is an exponential increase of embedded software in vehicles' ECUs. This stems from the increased complexity of the underlying control mechanisms but is also indicative of a trend of more development in software than in hardware, as shown in Fig. 4.2. This is possible because of tremendous improvement in the performance of microcontrollers. The advantage provided by this increase in software content is that software-based algorithms can be modified much easier, final parameterization can be done, and updates can be easily provided when necessary. The growth of software content, as shown in Fig. 4.2, has also increased the role of software engineering in mechatronics (Schäuffele and Zurawka 2016).
- *Intelligent Control:* Intelligent control provides adaptive and intelligent learning methods that supplement classical control techniques based on a mathematical model of the respective system. These techniques include a huge range of methodologies, from fuzzy controllers, neural networks, and expert-system-

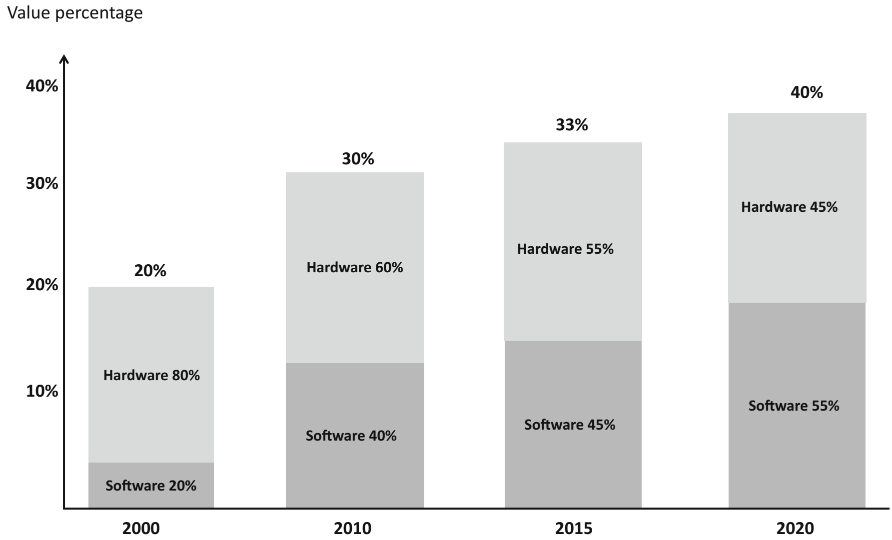


Fig. 4.2 Value percentage of Automotive E/E and Software in a vehicle (see also Mercer Management Consulting (2001) and (URL33 2017))

based control to hybrid and heterogeneous control techniques. Intelligent control is an important part of mechatronics as these systems often exhibit increased complexity. As the dynamics of a mechatronic system are often operating regime dependent, changes with external triggers sport a highly nonlinear behavior or need to adapt to various use case scenarios and nonstandard modeling techniques; and intelligent control methodologies become increasingly important.

- *Cyber-Physical Systems:* There is an increasing trend of connectivity in today's automotive systems which can be achieved by cyber-physical systems (see Sect. 5.1 in Chap. 5). These systems use computations and communication deeply embedded in and interacting with physical automotive systems by adding new capabilities to these systems. Cyber-physical systems (CPS) are networked systems which usually use open network technologies, such as the Internet. With the advent of the Internet of Things (IoT), sensors began to be networked, and mechatronics systems communicate with each other and backend systems. In this regard, computers and communication have become the universal system integrators that keep large systems together and which enable the composition of CPS infrastructure. Thus, CPS have an advanced and complex system architecture that connects computing, networking, and the physical and cyber, or virtual, environment within one paradigm, and therefore require a security design. CPS provide services such as:
  – Control
  – Information feedback
  – Real-time monitoring

  Most of the essential actions merge the interaction of the physical and the cyber worlds by integration and collaboration of computation, communication, and control (3C) (Ning 2013; Möller 2016). Such networked systems can be easily monitored remotely; however, they are also exposed to cybersecurity risks (see Chap. 6).

4.2 Automotive Electronics

Automotive electronics deals with any electrical system or component used in vehicles. The number of these systems has increased rapidly and continuously during recent years. On the one hand, many new sensors and actuators and, therefore, new specific ECUs (Sects. 4.2.5) have been developed to make drivers and passengers feel safer. Electronic control units are modular devices, and a modern vehicle can have from 45 to more than 120 embedded ECUs. On the other hand, entertainment and navigation systems have made their way into vehicles to make travel more comfortable.

Automotive E/E systems and components are distributed according to different automotive E/E application domains which can be divided into the major categories of:

- Body electronics
- Communication and entertainment systems
- Power control
- Safety control

At present, most body electronic products have entered the mature or recession period of the product life cycle. For the power controls category, the gross domestic income (GDI) is still growing; and its penetration rate in the European market is expected to increase to around 40% by 2018. With regard to the safety controls category, the assistance systems for safe driving are experiencing a rapid development and growth phase as they are moving from high-end cars to the mid-range. Future large-scale adoption depends on the maturity of the technology and a decline in costs. However, due to the lack of mandatory rules of law, it is still in the introduction period in emerging markets such as China. Recently, original equipment manufacturers (OEMs) have focused on new communications and entertainment systems.

Embedded in-vehicle information systems have seen rapid development as Europe and the USA require new cars to be equipped with emergency in-vehicle information systems. In-vehicle information systems will be more widely adopted in the mass market in the future. In general, automotive electronics can be divided into the following domains:

- Body electronics
- Chassis electronics

- Comfort electronics
- Driver assistance and advanced driver assistance electronics
- Electronic control units
- Entertainment/infotainment
- Sensor technology

The requirement for increased computing performance for automotive electronic devices is also accompanied by the driving need for high bandwidth in the vehicle network. In addition, a vast number of sensors, actuators, and motors in a multitude of vehicle control applications are being deployed.

Sensors in the automotive domain are essential to measure gases such as:

- COx
- NOx

They also measure:

- Speed
- Temperature
- Tire pressure
- Torque
- Vibration
- Yaw
- Other parameters helping to improve vehicle efficiency and safety

The actuators include:

- Drive pumps
- Electric motor drives controlled by engine control modules (ECMs)
- Fans
- Heating, ventilation, and air control (HVAC)
- Relays
- Solenoids
- Sunroofs
- Window lifts

All of these are essential to making vehicles more comfortable. The many electronic devices embedded in today's vehicles increase power consumption on one hand, based on the direct relationship of the weight of added hardware through the automotive electronic components and lower fuel economy. On the other hand, the need for more computing performance for these components, embedded memory capacity, and higher-bandwidth connectivity are each responsible for consuming additional power. In general, it can be stated that every 100 W of electrical power used requires approximately 0.1 l of gas/100 km or 0.1 l of gas/62 miles.

Every additional 50 kg of vehicle weight is responsible for approx. 0.15 l of gas/100 km or 0.15 l of gas/62 miles of increased fuel consumption. Thus, with the ever-growing levels of automotive electronic systems for comfort, efficiency, and safety, automakers and their electronic suppliers must cope with the conflicting demands of automotive electronics complexity versus vehicle power requirements and weight.

In addition to the greater computing and networking performance of automotive electronics, there is a major push for increased safety and security within the vehicle's body network to cope with the growing complexity of automotive electronics and the critical nature of the functions they enable.

- *Safety:* Safe comes from the Latin *salvus*, which means "uninjured" or "healthy." Safety studies and the practice of designing safe vehicles and equipment and complying with regulations all help with the goal of minimizing the occurrence and consequences of traffic collisions.
- *Security:* Secure comes from the Latin *securus*, which means "with care." Thus, security refers to a condition of being protected from or not exposed to danger. In the case of a vehicle, it offers collaborative opportunities for all stakeholders in the automotive industry to mitigate the risk of cyberattacks.

Nevertheless, the terms differ in connotation with regard to context when deciding which of them to use. However, as wireless communication for vehicles becomes more widespread, there is an ever-growing need for security in the automotive domain to prevent unauthorized access of the vehicle network for cyberattacks (see Chap. 6).

With regard to the advances in innovative automotive functionalities, the subtopics in automotive E/E can be summarized as shown in Fig. 4.3.

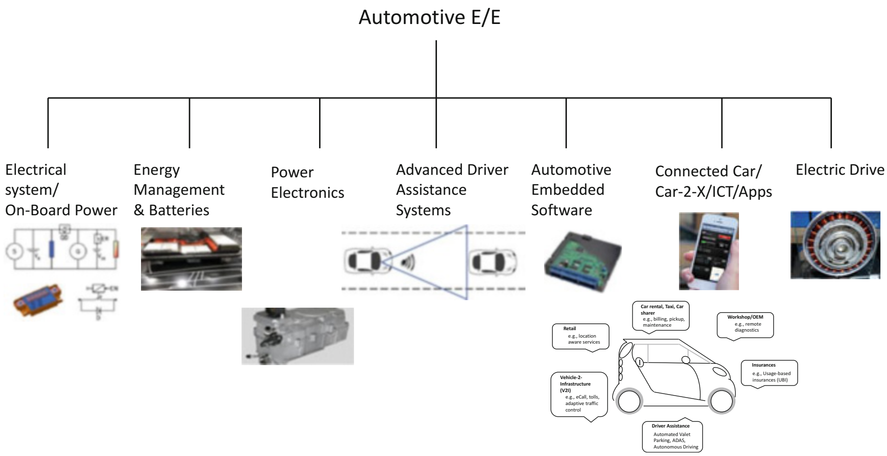


Fig. 4.3 Classification of subtopics in automotive E/E

4.2.1 Body Electronics

Body electronics are an important property for the acceptance of a vehicle with regard to functions such as:

- Central locking
- Lighting
- Window cleaning

Customers are very aware of all of these functions. Unfortunately, a well-engineered locking system does not persuade a customer to buy a particular vehicle brand; but a poorly designed one will definitely deter a customer from purchasing that brand. However, intuitive, logical, and clearly structured body electronic systems leave the customer with the impression of having a high-quality vehicle brand; but for the automaker, it is more difficult to develop systems with higher safety standards for an increasing number of vehicle variants and configurations. Hence, body electronics physical hardware is very important for customer because they are looking for new levels of comfort, efficiency, safety, and other features in their vehicle. Thus, body electronic systems are being developed for many different functionality, such as:

- *Body Control Module (BCM):* The BCM is responsible for monitoring and controlling various automotive electronic accessories in today's vehicle body. A BCM controls a number of devices, such as:
  - Air conditioning
  - Central locking
  - Immobilizer system
  - Power mirrors
  - Power windows

Furthermore, the BCM communicates with system ECUs through the vehicle's controller area network (CAN) or local interconnect network (LIN) bus (see Sect. 4.3.3). Its main application is controlling load drivers and actuating relays that in turn perform functions in the vehicle, such as locking the doors, dimming the in-vehicle overhead lamp, and other essential features (URL1 2017).

- *Power Management Module (PMM):* A vehicle PMM provides electronically switched power to the manifold E/E systems in a vehicle, such as data acquisition systems and automotive electronic devices, such as ECUs, lights, motors, solenoids, and others, and can be controlled through a combination of switch inputs, CAN messages (see Sects. 4.3.3 and 6.2.1), and logic functions. PMM reduces fuel consumption, maintenance costs, and repairs through:
  - Identification of degrading loads based on power usage or other parameters
  - Load prioritization/reconstitution/reduction/shedding
  - Power optimization of electrical loads or subsystems

Therefore, basic concepts of vehicle power management, as described in (Zhang and Mi 2011), include:

– Control device
– Data acquisition device
– Drive cycle selection device
– Fuel consumption and performance device
– Fuel demand in drive cycles device
– Monitoring device

- *Power Window and Door Control (PWDC):* The PWDC contains, within a vehicle's door, the automotive electronics used to drive all loads. It is connected via a LIN bus (see Sect. 4.3.3) to the dashboard, an easy-to-read, real-time user interface showing a graphical presentation of the current status and historical trends of the automotive electronic system, such as the vehicle's doors in conjunction with the vehicle's door module actuators, as shown in Table 4.1 for the ST Microelectronic microcontroller ST72F561, designed for midrange applications with CAN and LIN interfaces, based on an industry standard 8-bit microcontroller core, featuring an enhanced instruction set (ST Microelectronics 2013),

Table 4.1 Door module actuators

Actuators	
Door lock	1 DC motor
Mirror axis control	2 DC motors
Mirror defroster	1 DC motor
Mirror fold	1 grounded resistive load
Light bulbs	1 grounded resistive load
Window lift	1 DC motor

The following respective functions are realized through truth tables, an example of which is shown in Table 4.2 for the use case of door and window lock coding, as described in the *ST Microelectronics Application Note AN2334* (ST Microelectronics 2013):

– Light bulbs
– Mirrors
– Vehicle doors
– Windows

Vehicle doors are wired in many different ways, depending on which features are incorporated, such as the one that allows the driver to control all four windows on the vehicle and to lock out the controls on the other three individual windows. In this system, the power comes into the window switch control panel on the door and is distributed to a contact in the center of each of the four window switches. Two contacts, one on either side of the power contact, are connected to the vehicle ground and to the electric motor. The power also runs through the lockout switch to a similar window switch on each of the other doors.

Table 4.2 Window and door lock coding

Window				Door
Up left	Down left	Up right	Down right	Lock/Unlock
0	0	0	0	0
0	0	0	1	0
0	1	0	1	0
0	1	0	0	0
0	0	1	0	0
1	0	1	0	0
1	0	0	0	0
0	0	0	0	1
0	0	0	1	1
0	1	0	1	1
0	1	0	0	1
0	0	1	0	1
1	0	1	0	1
1	0	0	0	1

- *Remote Keyless Entry (RKE):* Automotive electronic remote central locking controls access to a vehicle and is activated by a handheld device performing the function of a standard car key without physical contact. RKE can include two functions: (i) remote keyless entry system which unlocks the doors and (ii) remote keyless ignition system (RKI) which starts the engine. The keyless entry system was originally a lock controlled by a keypad located at or near the driver's door that required pressing a predetermined numeric code for entry. These systems have evolved into a hidden touch-activated keypad today (URL2 2017).
- *Seat Comfort (SC)*: Smooth ride quality is an important issue, since a rough or bouncy ride can exacerbate back pain. Therefore, SC should focus on multicontour driver and front passenger seats which contain inflatable air chambers enabling them to adapt to the user's individual anatomy to give real orthopedic support. For the most part, these seats are developed by orthopedic specialists to provide outstanding comfort and support driver fitness, especially on long journeys.
- *Smart Mirrors and Wipers (SMW)*: For information on smart mirrors, see PWDC in this section. Smart wipers that activate themselves when it rains and adjust their speed when rain gets heavier or lighter weren't always commonplace. Rain can be sensed by a module mounted on the inside of the windshield, behind the rearview mirror. It contains light-emitting diodes (LEDs) and a set of light collectors. When the weather is dry, the LED light bounces off the windshield and into the collectors. When a raindrop comes down in front of the module, some of the light is refracted away from the collectors; and the system triggers the wiper blades to swipe.
- *Sunroof (SHD):* The SHD is controlled by the sunroof control module which contains all of the load circuits and is directly connected to the sunroof drive. The

module is allocated to the vehicle by means of encoding. It contains the following components:
– DC motor with attached step-down gear mechanism
– Two position-detecting hall effect sensors
– Electronic control switches

The two hall sensors register the number of DC motor revolutions and thus determine the position of the sunroof. The drive is switched off and on by reaching the relevant end position. The torque of the drive is constantly calculated from the pulses sent by the position sensors and the power consumption of the motor. Torque increasing to above a certain value is interpreted as trapping. Characteristic data for the antitrapping protection are defined in the coding data. The data are written into the control unit during the encoding procedure. Antitrapping protection is active in the close direction when the sunroof is open between > 4 and < 200 mm. This function is active during both normal closing and automatic operation and convenience closing of the sunroof. Antitrapping protection is deactivated in case of a fault by overpressing the switch in the close and hold direction. The closing procedure is interrupted if trapping is detected and the sunroof is opened for approximately 1 s (URL3 2017).

4.2.2 Chassis Electronics

The chassis system has a lot of subsystems which monitor various parameters and are actively controlled, such as:

- *Antilock Braking System (ABS)*: This safety system allows the vehicle's tires to maintain tractive contact with the road surface with regard to driver inputs while braking, preventing the wheels from locking up and avoiding uncontrolled sideslips. An ABS generally offers improved vehicle control and decreases stopping distances on dry and slippery surfaces; however, on loose gravel or snow-covered surfaces, ABS can significantly increase braking while still improving vehicle control. An ABS typically includes a central ECU, four wheel speed sensors, and two hydraulic valves within the brake hydraulics. The ECU constantly monitors the rotational speed of each wheel. If the ECU detects that a wheel is rotating significantly slower than the others, the ECU actuates the valves to reduce hydraulic pressure to the brake at the affected wheel, which reduces the braking force on that wheel and the wheel then turns faster. If the ECU detects that a wheel is turning significantly faster than the others, the brake hydraulic pressure to this wheel is increased so that the braking force is reapplied, slowing down the wheel. This process is repeated continuously. Thus, the wheels of cars equipped with ABS are practically impossible to lock even during panic braking in extreme conditions. Recent versions not only prevent wheel lock while braking but also electronically control the front-to-rear brake bias. This function, depending on its specific capabilities and implementation, is known as electronic brake force distribution (EBD) traction control, emergency brake assist (EBA), or electronic stability program (ESP) (URL4 2017).

- *Electronic Brake Distribution*: This brake technology automatically varies the amount of force applied to each of a vehicle's wheels, based on road conditions, speed, loading, etc. It can apply more or less braking pressure to each wheel in order to maximize stopping power while maintaining vehicular control. Typically, the front end carries the most weight, and EBD distributes less braking pressure to the rear brakes so that the rear brakes do not lock up and cause a skid. In some systems, EBD distributes more braking pressure at the rear brakes during initial brake application before the effects of weight transfer become apparent (URL5 2017).
- *Electronic Stability Control (ESC)/Electronic Stability Program:* This electronic system improves a vehicle's stability by detecting and reducing loss of traction. If ESP detects a loss of steering control, it automatically applies the brakes, helping to steer the vehicle where the driver intends to go. Braking is automatically applied to wheels individually, such as the outer front tire to counter oversteering or the inner rear wheel to counter understeering. Some ESP systems also reduce engine power until control is regained. ESP does not improve a vehicle's cornering performance; instead, it helps to minimize the loss of control (URL6 2017) (see also Sect. 4.2.5).
- *Traction Control System (TCS):* The TCS is a function of the ESP to prevent loss of traction. A TCS is activated when throttle input and engine torque are mismatched with regard to road surface conditions. The traction control system splits up the electrohydraulic brake actuator and wheel speed sensors with ABS.

Another type of automotive chassis electronic subsystem are the passive safety systems (PSSs). These systems are always ready to react when there is a collision in progress or to prevent a collision when it identifies a critical or dangerous situation. To these systems belong:

- *Airbag Control System (ACS)*: This safety system is designed to inflate the airbags very rapidly and then quickly deflate them during a collision in case of an impact with another object or a rapid sudden deceleration. The purpose of the ACS is to provide the driver and passengers with a soft cushioning and restraint during a crash event to prevent any large forces between the crashing driver and passengers and the interior of the vehicle. The airbag provides an energy-absorbing surface between the vehicle's driver and the steering wheel, the instrument panel, or the A-B-C structural body frame pillars, as well as the headliner and windshield/windscreen (URL7 2017).
- *Hill Descent Control (HDC):* The HDC system allows a smooth and controlled hill descent in rough terrain without the driver needing to touch the brake pedal. When the vehicle descends, the ABS system takes control of each wheel's speed. If the vehicle accelerates without driver input, the HDC will automatically apply the brakes to slow the vehicle down to the desired speed. With HDC, drivers can

be confident that even the ride down hills with slippery or rough terrain will be smooth and controlled and that they will be able to maintain control as long as sufficient traction exists (URL8 2017).

4.2.3 Comfort Electronics

Comfort electronics are automotive electronic systems that make a ride comfortable for the driver and the passengers like:

- *Automatic Climate Control*: Regulate cabin temperature and ventilation taking into account outdoor temperature, sun intensity, and cabin temperature with regard to the driver's or passengers' requests.
- *Electronic Seat Adjustment with Memory*: Keeps track of the user's seat position and stores the user's individual seat settings for the driver's seat and the front passenger's seat. Electronic seat adjustment with memory (ESAM) is available mostly in conjunction with memory exterior mirrors whose settings are also retrieved from the memory with regard to different drivers' or passengers' individual seat settings.
- *Automatic Wipers* (see Sect. 4.2.1): Activate themselves when it rains and adjust their speed when the rain gets heavier or lighter by detecting the amount of water on the windshield and controlling the wipers.
- *Automatic Headlamps—Adjusts Beam Automatically:* Activate through a photovoltaic sensor which is embedded into the instrument panel. The sensitivity of the sensor is either set by the automakers or the driver and is activated by the lighting conditions at dawn or dusk. The lights may switch off up to a couple of minutes after the engine has been turned off.
- *Automatic Cooling—Temperature Adjustment:* Maintain a constant temperature inside the vehicle. This requires the AC to regulate the inside temperature by an automatic control system using ambient air temperature sensors outside the passenger compartment. This type of sensor can be one or more in-vehicle air temperature sensors which also may include an infrared sensor that measures the actual body temperature of the driver and the passengers, a sunload sensor to compensate for sunlight entering the vehicle through the vehicle's windows, one or more electronic control modules, and electronic controls for the various heating, ventilation, and airflow control HVAC outlets. The controller keeps track of their position, running the motors full open and full closed and then counting the revolutions of the motor armature to determine their exact position (AA1Car 2016).

4.2.4 Driver Assistance Electronics

Driver assistance systems (DAS) are automotive electronic components developed to assist the driver in the driving process and to enhance vehicle systems for safety and better driving. Safety features are designed to avoid collisions and accidents by offering technologies that alert the driver to potential critical situations or to avoid collisions by implementing safeguards and taking over control of the vehicle, in effect bailing out the driver. The advanced driver assistance systems (ADAS) are a collection of systems and subsystems that finally result in a fully automated vehicle (see Sect. 4.9). The benefits of ADAS are potentially considerable because of a significant decrease in driver suffering, economic cost, and pollution. However, there are also potential problems to be expected, since the task of driving an ordinary vehicle is changing and moving in the direction of supervising an automated moving vehicle.

There are many different kinds of DAS automotive electronic components available. Some of them are built into vehicles or are available as add-on packages. A DAS relies on input from multiple data sources, including automotive imaging, computer vision, image processing, in-vehicle networking, light imaging detection and ranging (LiDAR), and radar. LiDAR is a surveying technology that measures distance by illuminating a target through laser light. Future autonomous vehicles will use LiDAR for obstacle detection and safe avoidance navigation through environments using rotating laser beams.

DAS, as well as ADAS, are fast-growing segments in automotive electronics, with steadily increasing rates of adoption of industry-wide quality standards in vehicular safety systems following the ISO 26262 for functional safety of automotive E/E systems defined by the International Organization for Standards (ISO) in 2011.

Next generation DAS will increasingly leverage wireless network connectivity to offer improved value by using vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) data using wireless fidelity (Wi-Fi®) data network systems (URL9 2017). V2V is a communication technology allowing vehicles to communicate with each other by Wi-Fi. V2V is also known as the vehicular ad hoc network (VANET), a variation of the mobile ad hoc network (MANET). V2I communication, commonly called vehicle-to-X (V2X), is the wireless exchange of critical safety and operational data between vehicles and road infrastructure intended to avoid or mitigate vehicle accidents but also to enable a wide range of safety, mobility, and environmental benefits.

Wi-Fi is the name of a wireless network technology that provides high-speed Internet and network connections based on the IEEE 802.11 standards. In addition, Wi-Fi is a registered trademark of the Wi-Fi Alliance, an organization made up of leading wireless equipment and software providers, certifying all 802.11-based products for interoperability, promoting the term as a global brand name across all marked 802.11-based wireless local area network (LAN) products.

Despite progress in using DAS over the past three decades, drunk driving claims a huge number of lives each year. Thus, the Driver Alcohol Detection System for Safety (DADSS) feature has been developed as an additional DAS measure and based on a technology which automatically detects when a driver is intoxicated with a blood alcohol concentration (BAC) at a certain breath alcohol level above a prescribed amount and prevents the vehicle from moving. There are two systems available for DADSS, the breath-based system and the touch-based system

(DADSS 2016). DADSS is supported by a broad coalition of organizations including automakers, safety and child advocates, bipartisan leaders in the US Congress and other government entities, and members of the medical community.

The most used DAS and ADAS components are:

- *Adaptive Cruise Control (ACC):* ACC is an intelligent form of cruise control that slows down and speeds up automatically to keep pace with the vehicle in front. The driver sets the driving speed; and the ACC radar sensor measures the speed ahead, monitors a vehicle in the lane, and instructs the following vehicle to stay a couple of seconds behind the vehicle ahead. The number of seconds to stay behind the vehicle ahead depends on the following driver's cruise control system settings. A schematic sketch is shown in Fig. 4.4.
- *Adaptive High Beam Assist (AHA)*: A headlight control strategy that continuously and automatically tailors the headlamp range so that the beam only reaches other vehicles ahead, AHA always ensures a maximum possible sight range without causing glare to other users on the road. The range of the beam can vary between 65 and 300 m, depending on traffic conditions.
- *Automated Parking Assist (APA):* APA is a maneuvering system that moves a vehicle from a traffic lane into a parking spot. The parking maneuver is achieved by means of coordinated control of the steering wheel angle and speed taking into account the actual surrounding environment using various sensor-based methods to detect objects around the vehicle. A signal is emitted that is reflected back when an obstacle is encountered near the vehicle, ensuring collision-free maneuvering within the available space.
- *Automotive Navigation System (see Sect. 4.2.6), Global Positioning System (GPS), and Traffic Message Channel (TMC)*: GPS is a US-owned utility that provides

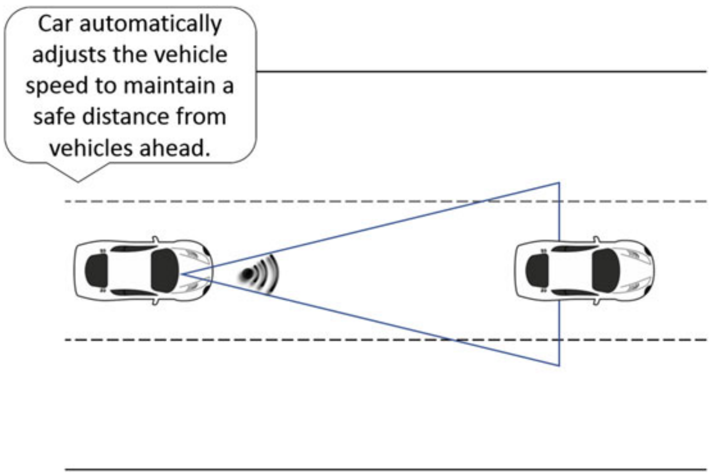


Fig. 4.4 Adaptive cruise control where the left vehicle in the bottom lane automatically follows the right vehicle at a safe distance

users with positioning, navigation, and timing services (PNT). It consists of three segments: space, control, and user. TMC delivers up-to-date traffic and travel information to vehicle drivers. It uses the ALERT C coding protocol to send messages via the Radio Data System (RDS) communication protocol standard embedding digital information in conventional FM radio broadcasts.

- *Collision Avoidance System (Pre-crash System):* Once an imminent crash is detected, the system either warns the driver that a collision is imminent or takes action autonomously without any driver input, e.g., by braking or steering or both. Collision avoidance by braking is appropriate at low vehicle speeds, e.g., below 50 km/h/ 31 mph, while collision avoidance by steering is appropriate at higher vehicle speeds.
- *Crosswind Stabilization (CS)*: Sensor-based systems detect forces acting on vehicles through side-wind gusts, such as when driving across a bridge or overtaking trucks. A crosswind stabilization system response takes into account:
  – Steering characteristics of the driver
  – Vehicle load
  – Vehicle speed

  Advanced crosswind stabilization regulates the suspension force according to the strength of the crosswind to reduce vehicle body oscillations.
- *Driver Drowsiness Detection (DDD):* This system helps prevent accidents caused by the driver getting drowsy. Various technologies are used:
  – Monitors the steering pattern using steering input from the electric power steering system
  – Monitors the vehicle's position in a lane using lane monitoring camera data
  – Monitors the driver's eyes/face by using a camera to detect the driver's face and eyeblink data
  – Uses body sensors to physiologically measure parameters such as brain activity, heart rate, skin conductance, and muscle activity
- *Electric Vehicle Warning Sounds for Hybrids and Electric Vehicles (EVWS)*: This system is designed to alert pedestrians to the presence of electric vehicles traveling at low speed. Warning sound devices are necessary because electric vehicles produce less noise than traditional combustion engine vehicles which can make it more difficult for pedestrians, the visually impaired, cyclists, and others to be aware of their presence. Warning sounds may be driver-activated electric warning systems or automatic systems at low speeds.
- *Emergency Assist (EA):* Emergency assist monitors driver behavior. In case of a medical emergency where the driver is no longer able to safely drive the vehicle, the vehicle takes control of the brakes and the steering until the vehicle comes to a complete stop.
- *Intersection Assistance (IA)*: The IA system monitors cross-traffic in an intersection/road junction. If the system detects a hazardous situation, it prompts the driver to start emergency braking by activating visual and acoustic warnings and automatically engaging the brakes.
- *Hill Descent Control (HDC)*: Hill descent control allows a smooth and controlled hill descent in rough terrain without the driver needing to touch the brake pedal (see Sect. 4.2.2).

- *Lane Departure Warning (LDW)*: This feature is designed to warn the driver when the vehicle begins to move out of its lane, unless a turn signal is on in that direction.
- *Parking Sensor (PS)*: A parking sensor system is designed to alert the driver to obstacles while parking.
- *Traffic Sign Recognition (TSR)*: This system is designed to recognize traffic signs on the road.
- *Vehicular Communication Systems (VCS)*: Vehicular communication systems are networks in which vehicles and roadside units are communication nodes providing each other with information, such as safety warnings and traffic information. VCS is part of intelligent transportation information systems (ITIS).
- *Wrong-Way Driving Warning (WWDW)*: This system emits an acoustic warning, together with a visual warning, which helps to prevent serious accidents caused by wrong-way drivers.

4.2.5 Electronic Control Units

An electric control unit (ECU) is basically made up of hardware and software where the software is a specific firmware based on the ECU's specific functionality. The hardware is mostly made up of various electronic components on a printed circuit board (PCB), with microcontroller chip(s) being the most important hardware component together with the following, depending on the ECU's properties:

- *Erasable Programmable Read-Only Memory (EPROM)*: Can retrieve stored data after its power supply has been turned off and back on; memory can be erased by exposure to a strong ultraviolet light source.
- *Flash Memory Chip*: Solid-state memory medium.
- *Electronic Solid-State Nonvolatile Storage Medium*: Can be electrically erased and reprogrammed.
- Other electronic hardware components.

The software (firmware) can be a set of lower- or higher-level codes that run the specific functionality on a microcontroller. An ECU can be characterized by its:

- Analog and/or digital inputs and outputs (I/O)
- Communication interface adapters
- Communication protocols
- Power device interface/control
- Switching matrices for low- and high-power signals

ECUs have different characterizing names based on their primary functionality. One of the most demanding ECUs in a vehicle is the engine control module (ECM). The main ECM function is to get information from sensors and, by running a certain actuator with the computed sensor information, to adjust their settings. This allows the vehicle to run in accordance with the driver's behavior. Connected sensors which play an important role in the ECM are:

- Absolute pressure sensor
- Air temperature sensor
- Camshaft sensor
- Crankshaft sensor
- Engine coolant temperature sensor
- Idle air controller (an actuator not a sensor)
- Knock sensor
- Mass air flow sensor
- Oxygen sensor
  - Zirconia oxygen sensor
  - Titania oxygen sensor
- Throttle position sensor

Because of the many sensors that measure/monitor the following in real time, as well as other parameters at different points within the engine, the ECM control demand is one with the highest real-time constraints, the so-called hard real-time requirement. Sensors measure/monitor:

- Engine speed
- Flow
- Nitrogen oxide (NO_x) level
- Oxygen level
- Pressure
- Temperature

All sensor information is sent to the ECM, which has logic circuits for doing the actual controlling. The ECM output is connected to different actuators, e.g., the throttle valve; the exhaust gas recirculation (EGR) valve; the fuel injector, which uses a pulse-width modulated (PWM) signal, dosing the injector; and more. Of all of the automotive electronics in any vehicle, the computing power of the ECM, typically a 32-bit microprocessor is among the highest.

Another important ECU is the transmission control module (TCM), which monitors/controls the transmission system, mainly gear shifting for better shift comfort, and lowers torque that is interrupted while shifting. An automatic transmission uses controls for its operation. Many semiautomatic transmissions also have fully or semiautomatic clutches. The ECM and TCM exchange messages, sensor signals, and control signals for their operation.

The vehicle control module (VCM) is connected to various kinds of sensors to monitor/control several systems in the vehicle. The VCM receives inputs from crash sensors (accelerometers) and sensors that detect the following to determine the force with which the frontal air bags should deploy:

- Seat belt use
- Seating position
- Seat position
- User's weight

Furthermore, the VCM take input from the following sensors to provide an output to the electronic stability control (ESC) for the safest driving situation.

- *Lateral Acceleration Sensors (LAS)*: Detect lateral acceleration of vehicles perpendicular to the direction of travel. It becomes noticeable as centrifugal force moves the vehicle to the outside of a curve when cornering. Lateral acceleration sensors are part of the electronic stabilization program.
- *Steering Wheel Angle Sensors (SWAS):* Measure steering wheel angle and velocity through the entire range of the steering wheel; merges optical and magnetic principles with advanced underlying software representing a mechatronic component. Vital part of the electronic stability control system.
- *Wheel Speed Sensors (WSS)*: Provide essential wheel speed information for ABS and for traction control and stability control.
- *Yaw-Rate Sensors (YRS)*: Gyroscopic device that measures a vehicle's angular velocity around the vehicle's vertical axis.

Hence, the main systems the VCM takes care of are:

- *Adaptive Cruise Control*: Automatically adjusts a vehicle's speed to maintain a safe distance from the vehicle ahead, see Sect. 4.2.4.
- *Airbag Control System*: Control system that detects and evaluates a crash before activating the appropriate restraint systems based on the type of collision and its severity, see Sect. 4.2.2.
- *Electronic Power Steering (EPS)*: Helps drivers to steer by augmenting the steering force of the steering wheel by using an actuator and a hydraulic cylinder that is part of a servo system. An EPS with no mechanical connection is called "steer-by-wire." In this context, wire refers to electrical cables that carry power and data, not thin-wire-rope mechanical control cables (URL10 2017).
- *Electronic Stability Control, Also Referred to as Electronic Stability Program*: Improves vehicle stability by detecting and reducing loss of traction (see Sect. 4.2.2).

4.2.6 Entertainment/Infotainment Electronics

Entertainment/infotainment systems are manufacturer specific, designed automotive electronic components for which different tools are used for hardware and software development. They are primarily developed by OEMs and/or third-party suppliers. The main types of entertainment/infotainment systems are:

- *Navigation Systems:* Are entirely on board a vehicle or located elsewhere and communicate via radio or other signals with a vehicle or use a combination of these methods:
  – Contain maps displayed in human readable format via text or in a graphical format
  – Determine vehicle's location via sensors, maps, or information from external sources
  – Provide suggested directions to a vehicle driver by text or voice
  – Provide directions directly to a connected car or an autonomous vehicle
  – Provide information on traffic conditions and suggest alternative directions
- *Vehicle Audio Systems:* Provide in-vehicle entertainment and information such as:
  – Navigation systems
  – Bluetooth® telephone integration
  – Smartphone controllers, such as CarPlay®, an Apple standard enabling a car radio or head unit to be a display and controller for an iPhone®, and Android™ Auto, a smartphone standard developed by Google allowing operation in vehicles through the dashboard's head unit

  Operated from the dashboard, these systems can be controlled by the steering wheel controls of the vehicle's audio system and simple voice commands to initiate phone calls, select radio stations, or play music from an MP3 player or other embedded devices.
- *In-Vehicle Infotainment (IVI)*: Hardware and software in vehicles that provide audio or video entertainment. In-vehicle entertainment originated with vehicle audio systems based on radio, cassette, and/or CD players and automotive navigation systems based on:
  – *Bluetooth and USB Connectivity:* Bluetooth was developed in the 1990s by the Bluetooth Special Interest Group as an industrial standard with regard to IEEE 802.15.1 for data transmission by radio waves between systems over a short distance. Bluetooth allows building up point-to-point, ad hoc, and piconet connections. A universal serial bus (USB) connects computers and peripherals and is an industrial standard developed by Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, and Philips. It provides a ubiquitous link that can be used across a wide range of PC-to-telephone interconnections.
  – *Carputer:* Computer with specific features, such as compact size, low-power requirement, and some customized components.
  – *In-Vehicle Internet*: Provided by tethering the Internet connection of a phone or tablet with other devices, such as laptops, or with a mobile hotspot, a physical location where the driver may obtain Internet access, whether portable or built into the vehicle.
  – *Video Player:* Hardware device to watch online video or view video files saved locally. Notable brands are Windows® Media Player and VLC media player.

– *Wi-Fi:* Technology that allows automotive electronic devices to connect to a wireless local area network (WLAN). A WLAN is usually password protected but may be open, which allows any device within its range to access the resources of the WLAN network.

Once controlled by simple dashboard knobs and dials, IVI systems can include steering wheel audio controls and hands-free voice control.

4.2.7 Sensor Technology

A sensor is a device that generates a measurable signal in response to a stimulus. It is capable of converting any physical quantity to be measured into a signal which can be displayed, read, stored, or used to control some other quantity of interest. Sensors are used to measure a particular characteristic of any kind of object or device. With regard to its application domain, a sensor is developed based on the type of use. For example, a thermocouple sensor, an electrical device consisting of two dissimilar conductors forming electrical junctions at differing temperatures, can be used to sense heat energy (temperature) at one of its junctions and generate an equivalent output signal, e.g., a voltage, which can be read by a multimeter.

Sensors are classified based on the nature of the quantity they measure which results in different kinds of sensors and, thus, different kinds of signals, e.g., electric, mechanical, optical, and others. A lot of sensors generate electrical signals (usually voltages) which are analogous to the physical quantity to be measured. Often, the voltage is proportional to the measurand quantity. Then, the sensor voltage output generated can be described by:

$$V_{\text{sensor}} = K \cdot m$$

where V_{sensor} is the sensor voltage output generated by the sensor, K is the sensitivity constant of the sensor, and m is the measurand. The sensitivity of a sensor is an important characteristic indicating how much the sensor's output changes when the sensor's input quantity being measured changes. It is basically the slope:

$$\frac{\Delta y}{\Delta x}$$

with Δy as sensor output and Δx as sensor input for a linear characteristic.

The typical static characteristics a sensor should have are:

- *High Accuracy:* Indicates the correctness of the sensor output in comparison to the actual measured quantity.
- *High Precision:* Represents the capacity of a sensor system which gives the same reading for respective measures of a measurand under the same conditions; closely related to high repeatability which means that the sensor system generates the same response for successive measurements when all operative and environmental conditions remain constant.

- High Resolution: Indicates the smallest change in the sensor input signal that the sensor can detect.
- *High Sensitivity:* Indicates the ratio of an incremental change in the sensor's output, Δy, to the incremental change of the measurand in the sensor input signal, Δx. For example, if the sensor output voltage of a temperature sensor changes by 1 mV for every 1 ^{o}C change in temperature, then the sensitivity is 1 mV/^{0}C.
  An ideal sensor has a large and preferably constant sensitivity in its operating range. The operating range describes the measurement range of the sensor representing the minimum and maximum values of the measurement that can be measured with the sensor system. Outside the measurement range, the sensor can, e.g., reach a saturation state at which it can no longer respond to any changes
- *Less Noise and Disturbance*: The noise refers to the signal-to-noise ratio (*S/N*) comparing the level of a desired sensor signal to the level of its background noise. *S/N* is defined as the ratio of signal power to noise power as described in the following equation:

$$S/N = \frac{P_{\text{signal}}}{P_{\text{noise}}}$$

with *P* as average power. An *S/N* higher than 1:1 indicates a higher sensor signal than background noise. Both signals must be measured at the same or the equivalent measuring point in the system under test and within the same system bandwidth.

So far, the variance of the sensor signal and background noise is known; and if the signal is zero mean, then the following equation can be obtained:

$$S/N = \frac{\sigma^2_{\text{signal}}}{\sigma^2_{\text{noise}}}$$

$$S/N = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}}\right)^2$$

If the sensor signal and the background noise are measured across the same impedance, then the *S/N* can be obtained by calculating the square of the amplitude ratio with *A* as root mean square (*RMS*) amplitude.

Disturbance refers to the impact to the measurement precision with regard to external or internal influences. Disturbance can be introduced as measurement error, defined as the difference between the true value of the quantity being measured and the actual value generated by the sensor. Hence, a measurement error can be defined as the real value of the sensor output of a measurement system minus the ideal value at the input of a measurement system according to:

$$e_{\text{ms}} = e_{\text{rum}} - e_{\text{itm}}$$

with e_{ms} as the measurement error, e_{rum} as the real untrue measurement value, and e_{itm} as the ideal true measurement value.

Since the measuring error can be caused by a variety of internal and external sources and is closely related to high accuracy, the absolute and relative error can be defined as follows:

$$\text{Absolute Error} = \text{Output} - \text{True Value}$$

$$\text{Relative Error} = \frac{\text{Output} - \text{True Error}}{\text{True Value}}$$

The absolute error has the same unit as the measured quantity; the relative error is unitless.

- *Less Power Consumption:* Minimized power source needs
- *Linearity:* Indicates that the sensor output signal changes linearly with the sensor input signal

Sensors are used to instrument and monitor a system or process, track assets through time and space, detect changes in the system or process which have been defined as being important, control a system or process with regard to being in closed vicinity within a defined range of change, and adapt services to improve their utility. Sensors are also used in everyday applications, such as touch-sensitive elevator buttons, lamps which dim or brighten by touching the base, and much more.

With advances in mechatronics and easy-to-use microcontroller platforms, the use of sensors has been expanded beyond the more traditional fields of measuring:

- Flow
- Pressure
- Temperature

Moreover, analog sensors, such as potentiometers, force-sensing resistors, and others are still widely used.

Sensors need to be designed to have little effect on the physical quantity measured, which requires that the sensor be manufactured to be very small and to consume less energy from the physical quantity measured to reduce measurement error. Sensors are hardware devices that range in scale from nano-sensors to macro-sensors. They act as data generators and pre- and/or post-processors of the data to be monitored. In the case of radio-frequency identification (RFID) sensors (see Sect. 5.2.2), the data processing is less complex as compared to other sensor types. The continuously produced analog signal x of the sensor is digitized into a proportional digital quantity by an analog-digital converter, and is sent to a microcontroller for further processing.

Sensors can be traditionally classified into the following categories:

- *Active Sensors:* Require continuous energy from a power source and sense data by actively probing the environment

- *Narrow-Beam Sensors:* Have a well-defined notion of the direction of measurement
- *Omnidirectional Sensors:* Have no notion of the direction involved in their measurements
- *Passive Sensors:* Self-powered and sense data without actually manipulating the environment by active probing

With the advent of new technologies, sensors are now manufactured on a microscopic scale, such as microelectromechanical systems (MEMS), a technology that in its most general form can be defined as miniaturized mechanical and electromechanical elements, devices, and structures, made using the techniques of microfabrication. They range in size from below 1 micron on the lower end of the dimensional spectrum to several millimeters and are fabricated as discrete devices or large arrays (Berlin and Gabriel 1997). MEMS devices can vary from relatively simple structures to extremely complex electromechanical systems under the control of integrated microelectronics. Moreover, MEMS perform two different types of functions: sensor and actuator. Both sensors and actuators act as transducers, converting one signal to another. Of specific interest are transducers that convert environmental information into digital signals and vice versa. Hence, MEMS sensors can convert environmental information, such as temperature, humidity, pressure, and more, into an electrical signal. In addition, MEMS actuators work in reverse to sensors; they convert an electrical signal into physical information to move or control devices, such as motors, hydraulic pistons, relays, and others. These MEMS components have high resonant frequencies leading to higher operating frequencies (Poslad 2009) and can be summarized as shown in Fig. 4.5 after (URL11 2017).

MEMS technology will enable a larger number of sensors to be used in connected cars and vehicles for autonomous driving, but this will require sophisticated networking to organize the real-time data transfer and fusion of the many sensors.

4.2.7.1 Sensor Nodes and Networks

Sensor nodes integrate sensors, actuators, computing elements, e.g., microcontrollers, memory, etc., communication systems; and a power source, e.g., a battery. A sensor node can also be a component of a larger network of sensors. Each sensor node in a sensor network is capable of performing processing, gathering sensor information, and communicating with other connected nodes in the network. They provide raw data to nodes responsible for sensor data fusion (SDF), which combines sensory data derived from distributed sources such that the resulting information has less uncertainty compared to the sources individually. Data sources

Fig. 4.5 Components of MEMS

MicroSensors	**MicroActuators**
MicroElectronics	**MicroStructures**

for sensor fusion are not necessarily specified to originate from identical sensors. Therefore, the sensor data fusion process can be:

- *Direct Fusion:* Based on a set of heterogeneous or homogeneous sensor outputs, on an algorithm where several measurements are processed together, or on historical values of sensor output data.
- *Indirect Fusion:* Uses information sources such as a priori knowledge about the environment and human input.

Raw data can also be processed by means of a sensor's computing capabilities, and the required output can be relayed to other sensor nodes.

Sensor networks usually interconnect many sensors or sensor nodes through wireless or wired connections (Golatowski et al. 2003). In data fusion in decentralized sensor networks with no central fusion system, no single sensor node has global knowledge of the network topology. Hence, algorithms are required which are able to locally process and assimilate data which finally yields a result identical to one obtained in a centralized system.

Wireless sensor networks (WSNs) consist of a large number of sensor nodes equipped with wireless network connections (WNCs) that can be deployed in the environment of the physical components of CPSs. In shared sensor and actuator networks (SANs), resource scheduling is an important feature for CPS operation.

The main components of a sensor node are, the microcontroller, the transceiver, the external memory, the power supply, and one or more sensors, resulting in a typical sensor node architecture as shown in Fig. 4.6.

The microcontroller component of the sensor node, shown in Fig. 4.6, performs specific tasks, processes data, and controls the functionality of other components in the sensor node. Microcontrollers are usually used because of their low cost, flexibility in connecting to other devices, easy programming, and low-power consumption.

Transceivers of sensor nodes represent a combination of a transmitter unit and a receiver unit into a single device. The operational states of transmitters are transmitting, receiving, idle, or sleep, which refers to their realization as state machines that

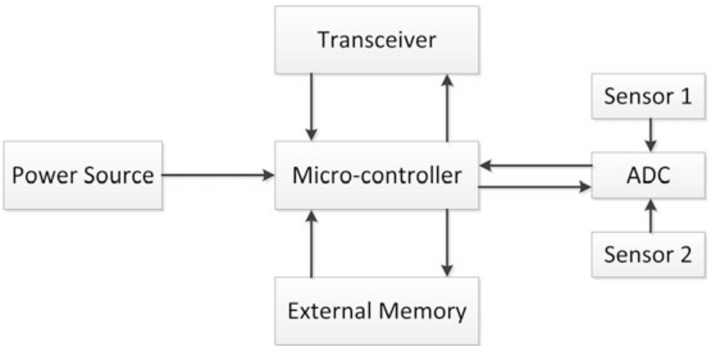


Fig. 4.6 Sensor node architecture

perform some operations automatically. A state machine is a mathematical model of computation used to design both computer programs and sequential logic circuits, acting on a set of inputs and computing a set of outputs. Thus, a finite state machine has a finite number of states to represent its state of processing. Its actions depend upon its internal state, and any inputs adhere to a specific syntax.

The memory requirements of sensor nodes depend on the specific application. There are two categories of memories usually used in sensor nodes:

- User memory to store application-related or personal data
- Program memory to program the device

The program memory can also contain the device's identification data.

An important issue in the development of wireless sensor nodes is ensuring that adequate energy is available to power the system because the sensor node requires power for sensing, communicating, and data processing. More of the required energy is used for data communication than for any other purpose. For example, the energy cost of transmitting 1 Kb a distance of 100 m (330 ft.) is approximately the same as that of executing 3 million instructions at 100 million instructions per second/W by a processor (URL12 2017). Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supplies in sensor nodes.

Wireless sensor nodes are typically very small electronic devices. They can only be equipped with a limited power source of less than 0.5–2.0 ampere hour and 1.2–3.7 volts (URL12 2017).

The energy efficiency for communication in a sensor network can be increased when using multi-hop topology (Zhao and Guibas 2004). In a N-hop network, overall transmission distance is Nd, where d is the average one-hop distance. The minimum receiving power is P_r, and the power at the transmission node is P_t. Thus, the power advantage P_A of a N-hop transmission versus a single-hop transmission over the same distance can be described as follows (Poslad 2009):

$$P_A = \frac{P_t(Nd)}{NP_t(d)} = \frac{(Nd)^a P_r}{Nd^a P_r} = \frac{N^a d^a P_r}{Nd^a P_r} = N^{a-1}$$

with a denoting the RF attenuation coefficient and $P_t \sim d^a P_r$.

Sensor nodes also play an important role in other domains, such as:

- Control systems
- Supervisory control and data acquisition (SCADA) systems
- Supervisory systems

For digital recording of analog quantities, analog-to-digital converters (ADC), shown in Fig. 4.7, are required (Möller 2003). The task of the ADC is to convert analog input variable X into a proportional output number. In many cases, time-dependent signals are digitized. For this purpose, the input quantity to be converted has to be sampled at a certain time and held. This task is performed by sample and

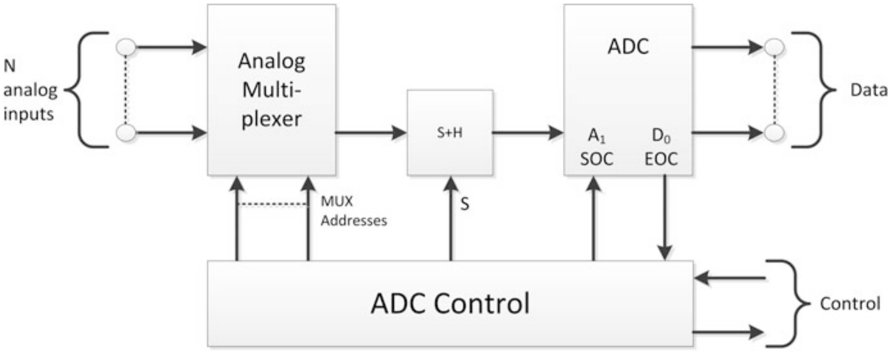


Fig. 4.7 Components of an analog-to-digital conversion system (Möller 2003)

hold (S/H) circuits, shown in Fig. 4.9. Very often, nonelectrical signals need to be digitally processed. Then, prior to the actual ADC conversion, the nonelectrical information needs to be converted into an electrical voltage. Sensors are used to detect nonelectrical information mapping an electrical voltage as output to the nonelectrical input.

In Fig. 4.7, the block structure of a conversion system converting analog inputs to digital data is shown. The control of the ADC is efficient for the following reasons:

- Several input channels are used; switching to a channel is controlled by using an analog multiplexer.
- After achieving the settling time, the S/H circuit switch is placed on hold; thus the conversion of a stable analog signal is possible. This does not affect the integration of the converter.
- Analog-to-digital conversion is started by the start-of-conversion (SOC) mode.
- After completion of the conversion, the analog-to-digital converter activates the end-of-conversion (EOC) mode.
- The ADC transfers the converted (digitized) measurements to the data processing unit (not shown in Fig. 4.7).

ADCs have a characteristic transmission curve in common, as shown in Fig. 4.8, with respect to the:

- Continuous abscissa pool y
- Discrete ordinate pool a

The intervals of variable Y can be mapped to a corresponding binary number a. In n-digit binary number $N = 2^n$, intervals are distinguished; the symmetrical ones are arranged around the abscissa values O, Y, $2Y \ldots$ $iY \ldots$ $(N\text{-}1)\,Y$, as shown in Fig. 4.8. Hence, the values of the input voltage are on average in accordance with the converted binary number.

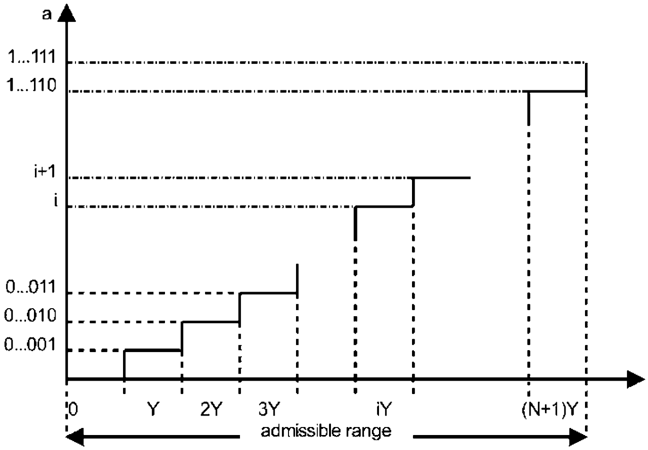
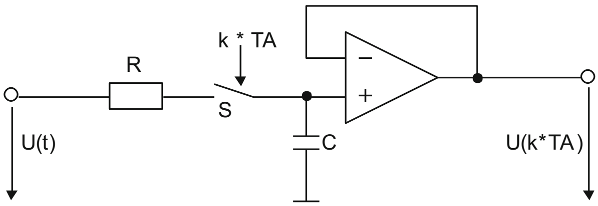


Fig. 4.8 ADC resolution characteristic (Möller 2003, 2016)

Fig. 4.9 S/H circuit of an ADC (Möller 2003, 2016)



Since the conversion process of analog-digital converters takes more than one clock cycle, the input signal U_{in} must remain constant throughout the conversion time period, which is achieved by an upstream S/H circuit. For this purpose, a hold signal is generated in the control logic of the converter, by which it is possible to define whether the input signal U_{in} should be held or should follow the real course of the analog input signal U_{in}. The S/H circuit consists of an operational amplifier connected as a voltage follower and a capacitor with low leakage current, as shown in Fig. 4.9.

4.3 E/E Architectures and Topologies

The number of electrical and electronic components in vehicles has increased rapidly and continuously during recent years. On the one hand, many new sensors and actuators and, therefore, new ECUs have been developed making passengers feel safer. On the other hand, entertainment and navigation systems have made their way into cars to make travel more comfortable.

The E/E architectures represent a consistent, vehicle-wide architecture of all E/E systems and E/E components on the hardware, software, electrical systems, cable assembly, and topology levels. They ensure that all components which are safe to operate are reliably and efficiently supplied with electrical energy. An innovative solution for E/E systems communication networking and control includes the adequate design of the E/E system, optimal integration of onboard electrical components, and intelligent energy management, as well as testing and validation of the entire E/E system. Thus, E/E architecture development has to be done with the following constraints:

- Modularization in vehicle development with the goal of scaling effects through the reuse of E/E system components or modules
- Protection of the E/E architecture in the concept phase with the goal of developing the E/E architecture prior to setting systems/components

Bus systems are the connecting part for communication purposes.

4.3.1 Objectives

E/E systems has soared these systems are increasingly important for efficient, comfortable, and safe vehicle systems functions. But the more technology that goes into a vehicle, the more important it is to optimize the E/E architecture. Additional functions, such as brake assistance, skidding control, parking aid, and others, have been embedded, which in turn leads to an increase in the number of ECUs and their communication links. Thus, developing E/E architectures is rigorous up-front work that involves the overall E/E system and subsystem design, the physical and functional partitioning, and the physical layout of subsystems within the vehicle. Therefore, E/E system development is the most important building block in designing the ever more complex technologies in automotive systems and products in a powerful and cost-effective manner.

With the respective system knowledge of E/E, networking, systems functions, and management, adequate conditions exist for successfully developing and implementing E/E systems and products. Starting with functions and the corresponding function chains, the design of components, data buses, and line connections exists and takes into account important boundary conditions, such as:

- Bus load
- Installation space
- Topology

As a result, the design of E/E architectures is a core requirement in the automotive domain. With regard to increasing complexity, many communication links and dependencies between composite functions prevent a simple decomposition of the E/E architecture design task into independent subtasks, which in turn consist of a number of individual components that communicate by signals with each other. Such components can either be sensors, processing components, actuators, or others.

Hence, E/E is a rapidly evolving field, driven by different requirement profiles, such as consumer demand, governmental regulation, increased E/E content, and others. Therefore, suitable networking systems are essential to exchange data quickly and securely. Examples are the bus systems, CAN, LIN, or FlexRay™, in the automotive field or the various Ethernet derivatives in the automotive industry, as well as in industrial and automation technology. Using these communication systems, the best possible interpretation is required. The transmission of all relevant data must also be ensured in critical situations in order to ensure the secure and perfect functionality of the overall system.

4.3.2 Architectures and Topologies

Modern automotive E/E architectures are growing in complexity to the point where it is difficult for the designer to predict the effects of the design decisions precisely. Thus, in addition to applying an architecture reference model to decompose the architecture, the design also requires domain-specific tools for synthesizing and evaluating the architecture during the design process. However, the complexity of E/E systems makes their optimization complicated, too, because multiple design goals refer to dependability and other non-functional requirements are crucial to take into account, such as:

- Adaptability
- Costs
- Maintainability
- Performance
- Power consumption
- Reliability
- Response time
- Safety

Based on a set of mechatronic subsystems with their sensors and actuators, a respective topology has to be defined in terms of data/signal integrity; however, several challenging issues have to be considered beyond the number of subsystems (ECUs). The required topology can be understood as a structure consisting of nodes and connections showing which nodes are interconnected. Different topologies are in use in vehicles that incorporate the E/E architecture-specific function into the vehicle, whereby its components can be assigned to ECUs. The ECUs, in turn, must be placed in an appropriate location in the vehicle and assigned with the respective hardware, depending on the components' requirements. Thus, ECUs have to be assigned to a topological structure such as:

- *Star Topology*: There is a central hub at which all ECUs are connected. Each ECU has its own line. If the central hub fails, the entire communication breaks down.

- *Bus Topology*: ECUs are connected by short branch lines to a main line. Every communication flows over this main line. If this main line is interrupted, two segments are formed which normally continue functioning. This topology is also called linear bus topology. Advantages of the linear bus topology are:
  - Bus topology is inexpensive.
  - The cable length required for this topology is the shortest compared to other networks.
  - It is easy to set up and to extend the bus network.
  - The linear bus network is used mostly in small networks.

  Disadvantages of the linear bus topology are:
  - A dependency on central cable has its disadvantages; if the main cable encounters a problem, the whole network breaks down.
  - It is difficult to detect and troubleshoot a fault at an individual station.
  - The efficiency of the bus network reduces as the number of devices connected to it increases.
  - The central cable length and the number of nodes that can be connected are limited.
  - Maintenance costs can get higher over time.
  - It is not suitable for networks with heavy traffic.
  - Termination is required to dump signals, and use of terminators is required.
  - Security, generally, is low because all of the computers receive signals sent from the source.
- *Ring Topology*: Point-to-point connection between ECUs is representative of ring topology. All connections are arranged in a closed chain. The communication can be done in only one direction. If a section of the line fails, the entire system no longer functions.

Figure 4.10 is an example of an abstract E/E architecture design introduced by (Moritz et al. 2011), in which gray circles with a black center point indicate sensors, gray circles with a white center point indicate actuators, the gray circle itself represents ECUs, and dashed lines refer to digital bus systems.

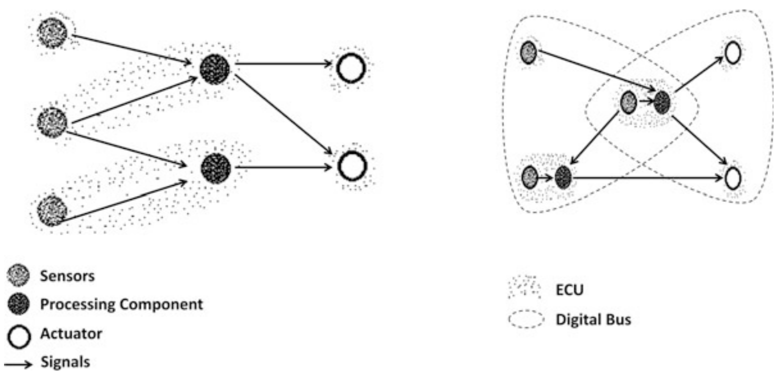


Fig. 4.10 E/E architecture design examples (Moritz et al. 2011) (for details, see text)

From Fig. 4.10, it can be seen that the E/E architecture of a given function contains several processing components, sensors, and actuators with corresponding ECUs assigned to digital buses, where the middle ECU acts as a gateway between the buses chosen. The arrows indicate the signal flow directions. The intrabus topology is not shown in Fig. 4.10.

The resulting architectures are evaluated according to two objectives, which are to be minimized (Moritz et al. 2011). The first one is cost, which is governed by cable and ECU cost. Cables in turn depend on the signals that have to be routed between the ECUs and on the communication structure used, as well as on the placement of the ECUs. The second objective is ECU complexity, defined as the average number of different functions assigned to an ECU. As ECU complexity decreases, the reliability of a vehicle, as to the number of functions which are affected by a single ECU failure, increases.

Optimization of the E/E architecture design can be based on the method of evolutionary algorithms which are primarily used for problem classes where classical standard methods fail or can hardly be applied (Yu and Gen 2010). In order to make the optimization procedure efficient, the evolutionary algorithm (EA) has to be adapted to the solution under test. Therefore, solution-specific knowledge is required for the evolutionary algorithm through (1) embedded local heuristics, (2) appropriate representation of solutions, and (3) corresponding variation operators. The use of local heuristics reduces the search space size of the evolutionary algorithm, but the available diversity of solutions in the population might be lost (Grosan and Abraham 2007). As described in (Moritz et al. 2011), the EA-optimized decisions are:

- Task 1: Assignment of components to ECUs
- Task 2: Physical placement of ECUs
- Task 3: Assignment of ECUs to digital buses

The decisions made by local heuristics are:

- Intelligent semiconductor selection for each ECU (simply take the cheapest one that satisfies the memory requirements of the processing components of that ECU).
- Bus type selection (choose the cheapest type that satisfies the data-rate requirements of the signals that have to be routed over the bus).
- Intrabus topology (choose such that the cable cost is minimized).

With regards to the aforementioned, the number of tasks that have to be optimized by the evolutionary algorithm has been reduced to the design of a suitable representation to optimize these tasks simultaneously. The assignment of components to ECUs corresponds to a partitioning of the components into clusters, whereby the number of clusters is the parameter to be minimized. The resulting E/E architectural representation of the hierarchical partitioning is shown in Fig. 4.11 (Moritz et al. 2011).

In Fig. 4.11, small gray circles indicate sensors or components; and small white circles indicate actuators. The arrows represent the data flow from sensors,

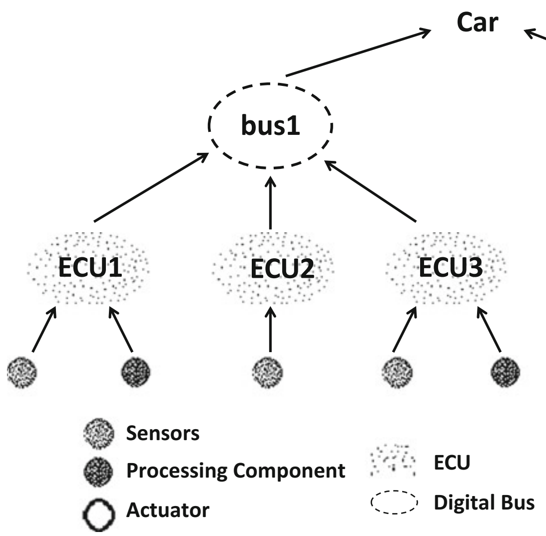


Fig. 4.11 E/E architecture design example representation after hierarchical partitioning (Moritz et al. 2011)

components, and actuators to ECUs, from ECUs to the bus systems, bus1 and bus2, and from the bus systems to the vehicle (car).

4.3.3 Bus Systems and ISO Standards

Modern vehicles have a large number of different kinds of ECUs through which specific vehicle functions are provided, as introduced in Sect. 4.2.5. One group of these ECUs can be stand-alone or distributed among the vehicles' EE multifunctional components. The majority are connected to one or more bus systems to control/monitor a broad range of vehicles. Another group of ECUs are those that have external interfaces, such as infotainment electronics, navigation systems, and others.

ECUs are interconnected through specialized internal communication networks inside vehicle bus systems. To meet the design challenges due to the different requirements, such as capacitance, real-time operation, and cost, several bus systems have been developed. The most important bus systems currently used in vehicles are:

- *Controller Area Network:* Inexpensive low-speed serial bus protocol for interconnecting automotive electronic components, such as microcontrollers and other devices, to communicate with each other in applications without a host computer, invented in 1986 by Robert Bosch GmbH and focused on safety, i.e., reliability. Bus nodes are all connected to the same shared bus line. A CAN bus is wired such that a 0-signal is dominant over a recessive 1-signal, as shown

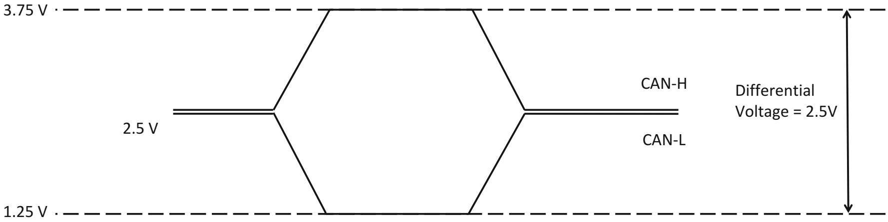


Fig. 4.12 CAN bus output signal

in Fig. 4.12. These dominant and recessive signals are used for a carrier-sense multiple access with collision avoidance (CSMA/CA) protocol that operates in the data link layer (layer 2) of the open systems interconnection (OSI) model that characterizes and standardizes the communication functions of a computing system without regard to their underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard protocols. The model partitions a communication system into seven layers. An arbitration scheme using priority resolution is used to decide which node is allowed to transmit data over the bus. The lower signal is its ID (and thus the more dominant 0s it sends during bus arbitration, the higher its priority). Using this scheme allows the CAN bus to communicate in real time.

Originally there was no built-in possibility to enforce security, such as encryption or authentication. Due to this original architecture, vehicles were vulnerable to cyberattacks, being hacked remotely and immobilized as a result of the hacking. The intrusion point for cyberattacks can be, for example, through the wireless tire pressure sensor (TPS) or directly, i.e., by physically accessing the CAN bus through the vehicle's entertainment/infotainment system. A solution to this problem is the CAN bus firewall (CBF) which separates each of the externally accessible ECUs from the shared bus by filtering the messages being sent from the ECUs and making sure that no hostile messages are going through and attacking the vehicle. It can also ensure that no unauthorized ECUs are connected to the vehicle. More details are discussed in Sects. 6.2.1 and 6.3.2.

- *Local Interconnect Network:* Broadcasting serial network protocol used for communication between components in vehicles comprised of 16 nodes. In the late 1990s, the LIN Consortium was founded by five automakers: Audi Group, BMW, Mercedes-Benz, Volvo, and VW Group, with networking technologies from Volcano Automotive Group and Motorola. The first fully implemented LIN specification was published in November 2002. The current LIN application combines the low-cost efficiency of LIN and simple sensors to create small networks. These subsystems can be connected by a backbone network, such as CAN, in vehicles.
- *FlexRay*: Automotive network communication protocol to govern onboard computing is a deterministic, error-tolerant, high-speed bus system developed by a consortium of leading automakers and Tier 1 suppliers with networking

technologies from National Instruments. It supports the essential needs required for drive-by-wire, steer-by-wire, and brake-by-wire applications. The consortium disbanded in 2009; but the FlexRay standard is now a set of ISO standards, ISO 17458-1–17458-5.
- *Ethernet for Control Automation Technology (EtherCAT):* Initiated by Beckhoff Automation as a real-time Ethernet, disclosed as International Electrotechnical Commission (IEC) standard 61158 protocols. EtherCAT is suitable for hard and soft real-time requirements in automation technology. The EthernetCAT Technology Group has 3905 members worldwide.
- *MOST Bus*: High-speed multimedia network technology optimized by the automotive industry and used for applications inside or outside the vehicle.

The standard multi master serial CAN bus (ISO 11898-1:2003) was originally specified as link layer protocol for the physical layer, for example, asserting the use of a medium with multiple access at the bit level through use of dominant and recessive states. The complexity of the ECU can range from a simple I/O device up to an embedded computer with a CAN interface and sophisticated software. The ECU may also be a gateway allowing a standard computer to communicate over a USB or Ethernet port to the devices on a CAN network. All ECU nodes are connected to each other through a twisted two-wire bus, CAN high (CAN-H) and CAN low (CAN-L). If a signal on the CAN-H wire goes from 2.5 to 3.75 V, the corresponding signal on the CAN-L wire goes from 2.5 to 1.25 V as shown in Fig. 4.12.

The CAN network, ISO 11898-2, is called a high-speed CAN using a linear bus terminated at each end with 120 Ω resistors. The electrical aspects of the physical layer, such as voltage, current, number of conductors, were specified by the International Organization for Standardization (ISO) as ISO 11898-2:2003. However, mechanical aspects of the physical layer, such as connector type and number, colors, labels, and pin-outs, have to be formally specified, too. As a result, an automotive ECU will typically have a particular connector with various sorts of cables, of which two are the CAN bus lines. The most common mechanical connector for the CAN bus is the 9-pin D-sub type male connector with the following pin-out:

- Pin 2: CAN-L (CAN−)
- Pin 3: GND (Ground)
- Pin 7: CAN-H (CAN+)
- Pin 9: CAN V+ (Power)

CAN is a serial bus protocol to connect individual sensors, ECUs, and systems as an alternative to conventional multiwire looms allowing automotive components to communicate on a single- or dual-wire networked data bus up to 1 megabit per second (Mbps). They are designed to allow microcontrollers and other devices (actuators, sensors, etc.) to communicate with each other in applications without a host computer. A CAN system allows the use of a single command station to control diagnostic systems and receive varied information, such as:

- Brake and transmission temperature
- Emissions levels
- Fuel efficiency
- Tire pressure

The CAN bus system is characterized by the following:

- All messages are broadcasted.
- Any node is allowed to broadcast a message.
- Each message contains an identification (ID) that identifies the source or content of a message.
- Each receiver decides to process or ignore each message.

There are four areas of application for serial communication (CAN) in automotive deployment which are subject to different requirements and objectives.

- *Connecting ECUs for Controlling Engine, Transmission, Suspension, and Brakes:* Data transfer rates are in the typical range for real-time applications ranging from 200 kilobit per second (kbps) to 1 megabit per second (Mbps).
- *Mobile Communication:* Connects components, such as vehicle radios, vehicle phones, navigation devices, etc., with a central ergonomically designed control unit.
- *Networking Components of Body and Convenience Electronics:* Multiplex applications, such as air control, air conditioning, central locking, and seat and mirror adjustment. Particular attention is paid to the cost of components and wiring. Typical data rates are in the vicinity of 50 kbps.
- *Onboard diagnostics (OBD)*: Hard-wired communication link to the ECU through which is allowed access to read and reset a vehicle's fault code. Fault codes are also known as diagnostic trouble codes typically made up of a letter followed by four numbers. So each code has a total of five characters, for example, B32XX. The first character, B, shows the identification for body systems, such as airbags, climate control, lighting, etc. The second character, 3, refers to a manufacturer-specific code, while the third character, 2, refers to the secondary air injection system. The fourth and fifth characters XX refer to the actual component that the ECU has identified with a fault. Once the OBD connector has access to the CAN bus, it is possible to monitor every component connected to it.

The CAN system bus data message structure, shown in Fig. 4.13, is the same for both the standard and the extended version.

SF	Message Identifier	Control	Data	CRC	ACK	EF
1bit	11 or 29 bits	6 bits	upto 64 bits	16 bits	2 bits	7 bits

Fig. 4.13 CAN bus data message structure

In Fig. 4.13, the acronyms have the following meaning:

- *Start Field* (SF): Marks the start of the data protocol. A bit with 3.75 V (depending on the system used) is sent over the CAN-H line, and a bit with 1.25 V is sent over the CAN-L line, i.e., the differential voltage is 2.5 V, as shown in Fig. 4.12.
- *Message Identifier:* Defines the priority level of the data protocol. If two CAN nodes try to transmit a message onto the CAN bus at the same time, the node with the highest priority (lowest arbitration ID) gets bus access. Depending on the standard being used, the length of the frames can be in two formats: standard, which uses an 11-bit arbitration ID, and extended, which uses a 29-bit arbitration ID, as indicated in Fig. 4.13
- *Control or Check Field:* Displays the number of items of information contained in the data field. This field allows any receiver to check if it has received all of the information transferred to it.
- *Data Field:* In this field, information is transferred to other CAN nodes.
- *Cyclic Redundancy Check or Safety Field (CRC):* Contains a 15-bit cyclic redundancy check code and a recessive delimiter bit. The CRC field is used to transfer fault detection.
- *Acknowledge Field or Confirmation Field (ACK):* Receivers send signal to transmitter that the data protocol has been correctly received. If an error is detected, the receivers notify the transmitter immediately. The transmitter then sends the data protocol again.
- *End Field (F):* Marks the end of the data protocol. The last possibility to indicate errors which lead to a repeat transfer.

CAN, LIN, and FlexRay are mainly used for control systems, whereas Media Oriented Systems Transport (MOST) is used for telematic applications. A bus system overview with regard to specific features is shown in Table 4.3.

Furthermore, intelligent and highly integrated actuator and sensor nodes in vehicles communicate via their adequate system network. Most body electronics applications use CAN or LIN communication interfaces. Application requirements, such as latency and bandwidth, as well as cost, influence the selection of a specific interface. Since the actual communication physical layer is driven mostly by electric and electromagnetic requirements, such as electromagnetic compatibility (EMC), electromagnetic interference (EMI), and electromagnetic discharge (EMD) standards, it is not a negligible portion of the device area. Normally, either a CAN physical layer or a LIN is integrated according to the respective application needs. In addition to CAN and LIN protocols, other communication interfaces like the ones for

Table 4.3 Bus system overview

	CAN	LIN	FlexRay	MOST
Application	Soft real-time systems	Low-level communication systems	Hard real-time systems	Multimedia, telematics
Bandwidth	500 kBit/s	19.6 kBit/s	10 Mbit/s	24.8 Mbit/s
Bus access	CSMA/CA	Polling	TDMA/ FTDMA	TDM/CSMA
Control	Multi master	Single master	Multi master	Timing master
Data bytes per frame	0–8	0–8	0–254	0–60
Physical Layer	Electrical (twisted pair)	Electrical (single wire)	Optical, electrical	Mainly optical
Redundant channel	Not supported	Not supported	Two channels	Not supported

the power train or chassis domain, such as Single-Edge Nibble Transmission (SENT) or Peripheral Sensor Interface 5 (PSI5), are gaining interest for use in further reducing network costs. For example, the use of PSI5 instead of LIN reduces the number of wires and connector pins from three (LIN, VBAT (battery voltage)), GND (ground)) down to only two (supply, GND). Even though PSI5 needs to modulate the data onto the supply line, the savings on the harness and connector side may be sufficient to account for the higher requirements on the E/E side.

In spite of the continued development of lower-cost protocols, the overall trend to CAN- and LIN-based nodes has been observed for many years in automotive sensor and actuator networks, especially in body electronics applications. According to strategic analytics, it is expected that in 2018, the number of CAN nodes will exceed the mark of $2 \cdot 10^9$; and the number of LIN nodes will exceed $1 \cdot 10^9$. In this regard, the average number of nodes per vehicle will be around 20 CAN nodes and approximately 10 LIN nodes. With a 17% compound annual growth rate, the expected market growth for LIN nodes is significantly higher compared to CAN nodes, with a 13% compound annual growth rate. This shows that simple functions are increasingly implemented in LIN nodes.

Ethernet, a family of computer technologies commonly used in LANs and metropolitan area networks, has been refined to support higher bit rates and longer link distances and will be the protocol of choice forming the backbone of the domain network. Also, a new super high-speed CAN bus called, CAN with Flexible Data-Rate (CAN FD), was developed by Bosch. CAN FD is positioned between the classic high-speed CAN and FlexRay and offers very high bandwidths. Many next generation ECUs will have a CAN FD interface. FlexRay and LIN will provide connectivity to intelligent nodes within a vehicle subdomain. But embedding powerful domain controllers will require an adequate support of this highly interconnected architecture. All cars sold in the USA since 1996 are required to have an onboard diagnostics (OBD) connector for access to vehicles' ECUs. Onboard diagnostics refers to a vehicle's self-diagnostic and reporting capability.

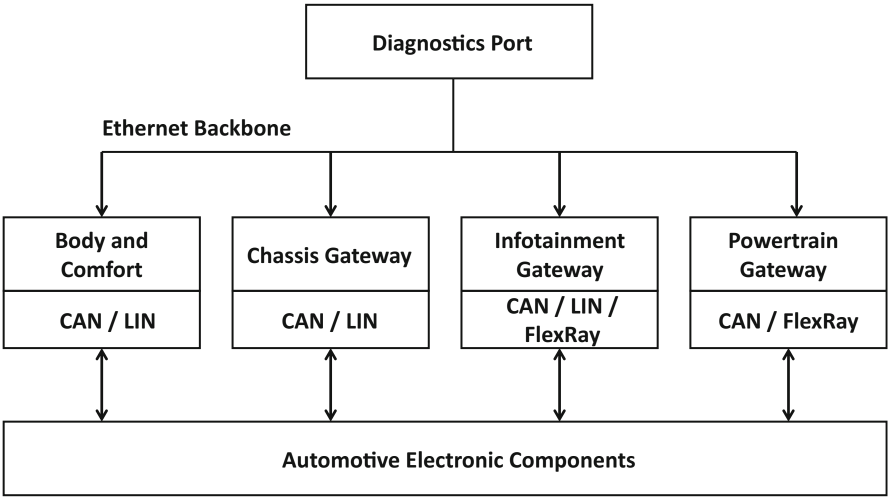


Fig. 4.14 Vehicle networks with regard to their application domains (URL1 2013)

In Fig. 4.14, an example of a vehicle network partitioned into separate application domains with associated domain controllers is shown. These domain controllers require significant amounts of processing power coupled with real-time performance and a plethora of communications peripherals.

The IEC 61508 standard, generally applicable to E/E programmable safety-related products, is only partially adequate for automotive electronics development requirements. Consequently, for the automotive industry, this standard is replaced by the existing ISO 26262, currently released as a Final Draft International Standard (FDIS). ISO/DIS 26262 describes the entire product life cycle (PLC) of safety-related EE components for vehicles. In this regard, PLC is the process of managing the entire life cycle of an automotive electronic component from inception, through engineering design and manufacturing, to service and disposal of manufactured components. It was published as an international standard in its final version in November 2011.

The implementation of this standard will result in modifications and various innovations in the automobile electronics development process, as it covers the complete product life cycle from the concept phase to decommissioning. It also has a safe assure solution, developed in accordance with the automotive functional safety standard (ISO 26262) and is targeted at specific safety functions of at least an Automotive Safety Integrity Level (ASIL) B rating. The safety integrity level (SIL) is defined as a relative level of risk reduction provided by a safety function or to specify a target level of risk reduction. The European functional safety standard is based on the IEC 61508 standard which defines four SILs where SIL 4 is the most dependable and SIL 1 is the least dependable, as shown in Table 4.4 (URL14 2017).

The SIL requirements for hardware safety integrity, shown in Table 4.4, are based on a probabilistic analysis of devices represented as a function of probability of failure on demand (PFD) and risk reduction factor (RRF) of low demand device

Table 4.4 SIL as defined in IEC 61508 (for details see text)

SIL	PFD	PFD (power)	RRF
1	0.1–0.01	10^{-1}–10^{-2}	10–100
2	0.01–0.001	10^{-2}–10^{-3}	100–1000
3	0.001–0.0001	10^{-3}–10^{-4}	1000–10.000
4	0.0001–0.00001	10^{-4}–10^{-5}	10.000–100.000

operation. The PFD is a measure of the effectiveness of a safety function, expressing the likelihood that the system will not perform the required safety function. For example, the likelihood that a SIL-3 system does not shut down a process when required is better than 1 in 1000 or 0.1%, as shown in Table 4.4. In other words, the availability of the safety function is better than 99.9%. Alternatively, it may help to think of a reduction of risk by a factor of 1000 (URL15 2017). The PFD of a one-channel system can be calculated by using a Markov model.

To achieve a given SIL, the hardware device must meet targets for the maximum probability of dangerous failure and a minimum safe failure fraction. Hence, a SIL is determined based on a number of quantitative factors in combination with qualitative factors, such as the development process and safety life cycle management.

4.4 Functional Safety

Functional safety is part of the overall safety of a vehicle system, or a component of it, that depends on the CPS and its components operating correctly in response to inputs, including the safe management of likely operator errors, hardware failures, and environmental changes. Functional safety is intrinsically end-to-end in scope as it has to treat the function of a system, subsystem, or component as part of the function of the whole system. This means that while functional safety standards focus on electrical, electronic, and programmable electronic (E/E/PE) systems, the end-to-end scope means that in practice functional safety methods have to extend to the non-E/E/PE parts of the system that the E/E/PE actuates, controls, or monitors. Therefore, the aim of functional safety is to bring risk down to a tolerable level and to reduce its negative impact; however, there is no such thing as zero risk. Functional safety measures risk by how likely it is that a given event will occur and how severe it would be or in other words, the amount of harm it could cause.

Thus, functional safety is achieved when every specified safety function is carried out and the level of performance required of each safety function is met. This is normally achieved by a process that includes the following steps as a minimum (URL1 2013); (URL13 2017).

- *Identify the Required Safety Functions:* This means hazards and safety functions have to be known or can be identified.
- *Assess the Risk Reduction Required by Safety Functions:* This involves a SIL, or performance level (PL), or other quantification assessment. A SIL applies to an end-to-end safety function of the safety-related system, not just to a component or part of the system.

- *Automotive Safety Integrity Level:* A risk classification scheme defined by the ISO 26262 standard, *Road Vehicles—Functional Safety*, which is an adaptation of the SIL used in IEC 61508 for the automotive industry. This classification helps define the safety requirements necessary to comply with the ISO 26262 standard. An ASIL is established by performing a risk analysis of a potential hazard by looking at the severity, exposure, and controllability of the vehicle operating scenario. The safety goal for that hazard, in turn, carries the ASIL requirements. There are four ASILs identified by the standard: ASIL A is comparable to SIL-1, ASIL B/C is comparable to SIL-2, and ASIL D is comparable to SIL-3. There is no ASL comparable to SIL-4. ASIL D dictates the highest integrity requirements for a product and ASIL A the lowest. However, ISO 26262 does not provide normative nor informative mapping of ASIL to SIL. ASIL is a qualitative measurement of risk; SIL is quantitatively defined as a probability or frequency of dangerous failures depending on the type of safety function. Thus, in IEC 61508, higher risk applications require greater robustness to dangerous failures.

  In general, IEC 61508-1:2010 covers aspects to be considered when E/E/PE systems are used to carry out safety functions. A major objective of this standard is to facilitate the development of product and application sector international standards by the technical committees responsible for the product or application sector. This will allow all of the relevant factors associated with the product or application to be fully taken into account and thereby meet the specific needs of users of the product and the application sector. A second objective of this standard is to enable the development of E/E/PE safety-related systems where product or application sector international standards do not exist.

  Furthermore, IEC 61508-3:2010 applies to any software forming part of a safety-related system or used to develop a safety-related system within the scope of IEC 61508-1 and IEC 61508-2. It provides:

  – Specific requirements applicable to support tools used to develop and configure a safety-related system within the scope of IEC 61508-1 and IEC 61508-2
  – Requires that the software safety functions and software systematic capability are specified
  – Establishes requirements for safety life cycle phases and activities which will be applied during the design and development of the safety-related software

  These requirements include:

  – Applying measures and techniques which are graded against the required systematic capability for the avoidance of and control of faults and failures in the software.
  – Providing requirements for information relating to the software aspects of system safety validation to be passed to the organization carrying out the E/E/PE system integration
  – Providing requirements for the preparation of information and procedures concerning software needed by the user for the operation and maintenance of the E/E/PE safety-related system
  – Providing requirements to be met by the organization carrying out modifications to safety-related software

– Providing, in conjunction with IEC 61508-1 and IEC 61508-2, requirements for support tools, such as development and design tools, language translators, testing and debugging tools, and configuration management tools

The second edition cancels and replaces the first edition published in 1998. This edition constitutes a technical revision. It has been subject to a thorough review and incorporates many comments received at the various revision stages (URL16 2017).

- *Ensure Safety Function Performs to the Design Intent:* This includes under conditions of incorrect operator input and failure modes. It also involves having the design and life cycle managed by qualified and competent engineers carrying out processes to comply with a recognized functional safety standard. In Europe, that standard is IEC EN 61508, one of the industry-specific standards derived from IEC EN 61508, or some other standard, such as ISO 13849.
- *Verify That the System Meets the Assigned SIL* (ASIL, PL, or Agricultural Performance Level (agPL))*:* This can be accomplished by determining the mean time between failures (MTBF) and the safe failure fraction (SFF), along with appropriate tests. SFF is the probability of the system failing in a safe state. The critical or dangerous state is identified from a failure mode and effects analysis (FMEA) or failure, mode and effects and critical analysis (FMECA) of the system under test.
  – *Mean Time Between Failure (MTBF):* The predicted elapsed time between inherent failures of a system during operation which can be calculated as the arithmetic mean time between failures of a system using the following equation:

$$\text{MTBF} = \frac{\sum(\text{start of downtime} - \text{start of uptime})}{\text{number of failures}}$$

  – *Mean Time to Dangerous Failure ($MTTF_d$):* The $MTTF_d$-value should primarily be provided by the system manufacturer. If the manufacturer cannot provide the required values, they can be taken from ISO 13849-1 tables or can be calculated using the B_{10d}-value, (average number of cycles until 10% of the components have a dangerous failure). To calculate the $MTTF_d$, it is also important to know the average number of cycles per year the component will execute.

  A $B_{10d} = 2 \cdot 10^6$ results in $MTTF_d = 1{,}141$ year which corresponds to the level $MTTF_d$ = high.

$$\text{MTTF}_d = \frac{B_{10d}}{0.1 \cdot n_{op}}$$

where

$$n_{op} = \frac{d_{op} \cdot h_{op} \cdot 3600}{t_{cycle}}$$

with

$$
\begin{aligned}
n_{\text{op}} &= \text{Number of cycles per year} \\
d_{\text{op}} &= \text{Operation days per year} \\
h_{\text{op}} &= \text{Operation hours per day} \\
t_{\text{cycle}} &= \text{Cycle time in seconds}
\end{aligned}
$$

– *Safe Failure Fraction:* Takes into account any inherent tendency to fail toward a safe state. An SFF is the sum of the rate of safe failures plus the rate of detected dangerous failures divided by the overall failure rate:

$$\text{SFF} = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D}$$

It is important to realize that the only types of failures to be considered are those which could have some effect on the safety function.
where

$$\lambda_S : \text{Rate of safe failure}$$

$$\left(\sum \lambda_S + \sum \lambda_D \right) : \text{Overall failure rate}$$

$$\lambda_{DD} : \text{Rate of detected dangerous failure}$$

$$\lambda_D : \text{Rate of dangerous failure}$$

– *Failure Mode and Effects Analysis (FMEA):* The first step of a system reliability study involving reviewing as many components, assemblies, and subsystems as possible to identify failure modes and their causes and effects. For each component, the failure modes and their resulting effects on the rest of the system are recorded in a specific FMEA worksheet. FMEA can be a qualitative analysis but may be put on a quantitative basis when mathematical failure rate models are combined with a statistical failure mode ratio database (URL17 2017).
By early dealing with possible sources of error, a strategy of error avoidance is followed instead of elaborate error correction. The FMEA is, therefore, particularly suitable for new developments and changes to products and processes. The risk assessment enables critical components to be found and priorities to be set in the prevention of errors.
– *Failure Mode and Effects and Criticality Analysis (FMECA)*: Extended FMEA indicates that criticality analysis is also performed. For each failure mode, the ability of the system to detect and report the failure in question is analyzed. One of the following will be entered on each row of the FMECA matrix:
  • *Normal*: The system correctly indicates a safe condition to the designer.

- *Abnormal*: The system correctly indicates a malfunction requiring design action.
- *Incorrect*: The system erroneously indicates a safe condition in the event of malfunction or alerts the designer to a malfunction that does not exist (false alarm).

  Failure mode criticality assessment may be qualitative or quantitative. For qualitative assessment, a mishap probability code or number is assigned and entered on the matrix. For example, MIL-STD-882 uses five probability levels:

Level	Description
A	Frequent
B	Probably
C	Occasional
D	Remote
E	Improbably

The criticality numbers are computed as:

$$C_m = \lambda_p \frac{\alpha}{\beta \cdot t}$$

and

$$C_\tau = \sum_{n=1}^{N} (C_m)_n$$

with

$$\lambda_p = \text{Basic failure rate}$$
$$\alpha = \text{Failure mode ratio}$$
$$\beta = \text{Conditional probability}$$
$$t = \text{Mission phase duration}$$

- *Conduct Functional Safety Audits*: Examine and assess the evidence that the appropriate safety life cycle management techniques were applied consistently and thoroughly in the relevant life cycle stages.

Neither safety nor functional safety can be determined without considering the vehicle cyber-physical system as a whole and the environment with which it interacts. Functional safety is inherently end-to-end in scope.

4.5 Automotive Software Engineering

The automotive industry faces global competition where speed, cost-efficiency, and innovative power are decisive factors for securing the future of automakers. Current challenges for automakers, such as innovative drive concepts, Car IT, and driver assistance systems increase the demand for automotive software engineering with regard to information technology (IT) and E/E. This trend is being reinforced by the networking of vehicles for assistance and piloting services.

The key challenge today is that almost all functions of a vehicle are electronically controlled or monitored. The realization of vehicle functions controlled by software offers unique degrees of freedom in the design process. But the main difference between automotive software and other types of software, such as personal computers and telecommunication systems, is in the essential requirements for reliability. In a complex ECU network, automotive software must be exceptionally reliable over the full vehicle life cycle. Therefore, in vehicle development, main boundary conditions, such as high reliability and functional safetyrequirements, comparatively long product life cycles, complexity of software functions, limited costs, shorter development times due to shorter innovation cycles, and an increasing variety of vehicle variants, have to be considered. The number of software functions of a motor control unit has now reached the three-digit range.

However, powerful software functions must be implemented in vehicles which interact internally but also have numerous interfaces to functions in the chassis or body area, for example, to the drive slip control or automatic climate control, and other features. Typical is the high number of parameters, such as characteristic values, characteristic curves, and key fields, which are used to coordinate the software functions for the respective systems' or components' undisturbed functionality, such as engine, gearbox, vehicle variant, and other important functions (Schäuffele and Zurawka 2016).

Another essential requirement for automotive software engineering is that the ECUs are highly interconnected by in-vehicle networks. ECUs communicate via standardized bus systems, such as CAN, LIN, FlexRay, and MOST. In contrast to Ethernet, well-known from PC connectivity, CAN and LIN bus systems are rather slow. In vehicles information must be processed within milliseconds. However, the increasing number of connected ECUs has led to more elaborate structures. Thus, in-vehicle networks require specific E/E architecture designs. To master the complexity of these networks, vehicle ECUs is partitioned into domains, such as power train, chassis, body/interior, infotainment, and others, whereby each domain has different requirements. For example, the power train domain requires extremely precise timing, closed-loop control, and real-time behavior, whereas infotainment needs optimal presentation of information. For example, the power train domain requires extremely precise timing, closed-loop control, and real-time behavior, whereas infotainment needs optimal presentation of information.

4.5.1 Increasing Software Content and Product Complexity

Today's E/E systems carry out many functions in modern vehicles, including driver assistance functions, vehicle dynamics control, active/passive safety systems, and other features, the functionality of which is embedded through software content. Automotive software has several main requirements with regard to:

- *Functional Safety*: Functions such as antilock braking, ESC, and others require fail-safe operation, which puts high demands on software development processes and the software functionality itself.
- *Minimized Resource Consumption*: Additional computational power and memory capacity must be minimized as the need for them continues to grow.
- *Real-Time Behavior*: Defined fast reaction on external incidents requires optimized operating systems and specific software content design.
- *Reliability:* Automotive software must be exceptionally reliable within the complex and multifaceted ECU networks over the full vehicle life cycle.

These requirements result in increased complexity of ECU software functions and therefore increased product complexity with regard to the number of lines of code.

The methods, standards, and processes used for automotive software development and tests are:

- *Agile Software Development*: Principles for software development based on which requirements and solutions evolve through the collaborative effort of self-organizing, cross-functional developer teams.
- *Automotive SPICE®*: Industry-specific standard derived from ISO 15504 for software process assessments, published by the Special Interest Group Automotive in 2005. It has two dimensions: its own process reference model (PRM) and process assessment model (PAM).
- *AUTOSAR*: A worldwide development partnership founded in 2003 (see Sect. 4.6).
- *Diagnostic Standard of Association for Standardization of Automation and Manufacturing (ASAM) Systems:* Provides standards for data models, interfaces, and syntax specifications for a great number of applications, examples, evaluations, and simulations.
- *HIL:* Technique used in developing and testing complex real-time systems by using simulation (see Sect. 4.5.3).
- *Model-Based Development:* Model-based method addressing problems associated with developing complex control systems (see Sect. 4.5.2).
- *Safety Standards*: According to ISO 26262 and IEC 61508.
- *State-of-the-Art Requirements Engineering*.

The standards used for network connection are:

- CAN (see Sect. 4.3.3)
- Ethernet (see Sect. 4.3.3)
- FlexRay (see Sect. 4.3.3)
- LIN (see Sect. 4.3.3)
- MOST (see Sect. 4.3.3)

The increasing complexity of software applications requires efficient development of high-quality software code. Thus, software engineering, can be regarded as a study and application of engineering to the design, development, and maintenance of software, becomes very important. In this regard, agile software development describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement; and it encourages rapid and flexible response to change. These principles support the definition and continuing evolution of many software development methods.

Twelve principles of agile software development (ASD) have been defined by the Agile Alliance to supplement the Manifesto for agile software development (see Sect. 7.2). They are as follows (Holtz and Möller 2017):

- The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective and then tunes and adjusts its behavior accordingly.

Well-known ASD methods include (Holtz and Möller 2017):

- Agile modeling
- Agile unified process (AUP)
- Dynamic systems development method (DSDM)
- Essential unified process (EssUP)
- Extreme programming (XP)
- Feature-driven development (FDD)
- Open unified process (OpenUP)
- Scrum
- Velocity tracking

Automotive SPICE is a framework established in ECU development to improve and evaluate processes. The industry-specific standard was developed in 2005 based on ISO/IEC 15504 for software process assessments and adapted for requirements in the automotive sector. Automotive SPICE® V3.0 was presented to the public at the VDA Automotive Systems 2015 conference as a new standard. With this revision, Working Group 13, commissioned by the German Association of the Automotive Industry, ensured that Automotive SPICE is in harmony with the new ISO/IEC 12207 and ISO 15504-5. Therefore, in version 3.0, the previous engineering processes are divided into system and software processes. In addition, there are many improvements in the model. The German automotive manufacturers, Audi, BMW, Daimler, and Volkswagen, have agreed on a minimum subset of 16 processes (Hersteller Initiative Software or HIS Scope), which are assessed by each member in the software initiative. The HIS Scope of Automotive SPICE, as the minimum requirement of the processes to be considered in an assessment, is also used in other branches of industry as a starting point for process improvement and a focus for assessments.

ASAM MCD-2 (Association for Standardization of Automation and Measuring Systems) allows the data-oriented specification of vehicle diagnostics. The standard defines a data model for the description of diagnostics capabilities of ECUs needed throughout the life cycle of a vehicle from development to testing, production, aftersales, and service. The standard facilitates the exchange of diagnostic information between partners in the development process, e.g., between OEM and Tier 1 suppliers or between OEMs in a cooperation project. In detail, ASAM MCD-2 covers the description of:

- Diagnostic communication via requests and responses, diagnostic trouble codes, parameters, and other diagnostic data
- Communication parameters for different diagnostic protocols
- ECU memory programming
- ECU variant coding
- Function-oriented diagnostics

The standard defines a data model and description format which is independent of specific vendors, buses, or protocols and which has well-defined semantics for all specification elements. ODX allows the user to store diagnostic data in a central location and to efficiently distribute the data to all involved parties from a single source. ODX data is serialized in machine-readable Extended Markup Language (XML) format. Therefore, the standard enables the complete reuse of diagnostic data throughout all development phases of an ECU, e.g., for the design of diagnostic communication, the development of the ECU kernel and application software, configuration of diagnostic testers, and the generation of the diagnostics documentation for the vehicle. Since the data originates from one source, ODX helps to prevent inconsistencies, errors, and repetitive efforts (URL18 2017).

4.5.2 Model-Based Development

The complexity of automotive systems operations, especially safety functions, makes predicting safety performance extremely difficult in the product development process, which can be supported by the model-based development approach shown in Fig. 4.15. This approach focuses on the model of the vehicle systems and its corresponding control algorithms. A lot of care has to be taken to capture all relevant requirements and features of the vehicle systems, and extensive simulations are done to validate the overall closed control loop system functionality.

In a classical, nonmodel-based development scenario, the parameter of the controller defines the variables and constants for a piece of C-code that implements the control algorithm for the chosen target architecture. In model-based development, this step is automated. The dynamic model of the controller provides the input to software called an auto-code generator that directly produces the target C-code.

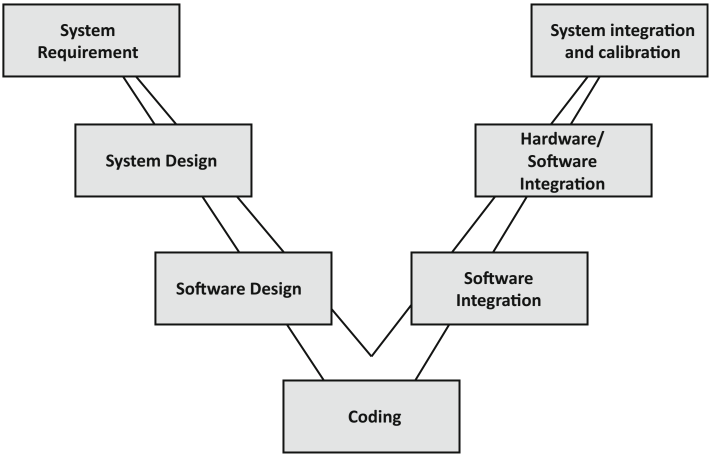


Fig. 4.15 Model-based design process

As the auto-code generator has been tested extensively, the resulting code is of excellent quality and minimizes standard programming errors, such as buffer overflow and others, which are often found in manually generated code.

- *Algorithm Design:* Simulation using tools such as MATLAB®/Simulink®, Advanced Simulation and Control Engineering Tool (ASCET), and LabVIEW
  - *MATLAB/Simulink*: MATLAB is the core product for numerical calculations. It is used in almost all industries and sciences for calculation and data evaluation. Simulink is the quasi-standard for the modeling of dynamic systems.
  - *ASCET*: ASCET provides software tools needed to successfully develop model-based application software and generate C-code. The notations, which ASCET uses for modeling, allow engineers to capture application software designs quickly and effectively. Through a combination of static analysis and testing, designs can be validated efficiently early in the development life cycle. The tool also supports model-based development, validation, measurement, and calibration of AUTOSAR-based ECU software.
  - *LabVIEW*: A graphical programming platform supporting engineers in scaling from design to test and from small to large systems. Integrating all tools that engineers need to build a wide range of applications in less time, LabVIEW is a development environment for problem solving, accelerated productivity, and continuous innovation.
- *Modeling and Simulation, Auto-Code Generation*: As a result of automated code generation technology, another application of these models has become viable which can serve as input to an automatic embedded code generation process. The code generated from these models is highly efficient, readable, testable, and suitable for use in safety-critical applications.
- *Rapid Prototyping With Porting Optimization:* A method that helps in validating numerical analysis results and shortening the development cycle, whereby it is necessary to choose the optimization parameters, which usually refer to cost, for each optimization cycle.
- *Model-in-the-Loop Testing:* The first representative of in-the-loop tests. For this test, a model of the ECU environment to be developed is required. The environmental model can have a high degree of abstraction, since, for example, sensors and actuators do not have to be completely modeled, their input and output behavior can be directly implemented.

Model-based development is a proven approach for efficiently developing solutions to complex engineering problems. It is a method for developing complex systems using mathematical models of system components and their interactions with the surrounding environment. These models have many applications in the development process, including system simulation, stability analysis, and control algorithm design specific to ECUs. Several off-the-shelf software tools for model-based development support the specification, design, and validation of high-reliable

ECU models for a wide range of system design in automotive applications. The resulting models can be as detailed as necessary, assuming that sufficient information is available to support the construction and validation of the model. One important application for high-fidelity ECU models is simulation. A model of the ECU for a high-fidelity vehicle is developed and integrated into the vehicle system simulation. By working with a system-level simulation that combines the vehicle and its respective ECU, it is possible to thoroughly test the control system design and rapidly make changes and improvements as needed.

Model-based development takes into account ISO 26262 a functional safety standard intended to be applied to the development of software for E/E systems in the automotive domain. ISO 26262 is an adaptation of the broader IEC 61508 safety standard. ISO 26262 provides an automotive safety life cycle, including management, development, production, operation, service, and decommissioning, and supports tailoring the necessary activities during these life cycle phases. Furthermore, it covers functional safety aspects of the entire development process, including such activities as requirements specification, design, implementation, integration, verification, validation, and configuration. ISO 26262 also provides an automotive-specific, risk-based approach for determining ASIL risk classes. It uses ASIL to specify an item's necessary safety requirements for achieving acceptable residual risk and provides requirements for validation and confirmation measures to ensure that a sufficient and acceptable level of safety is being achieved.

Today's software development, however, has big challenges in balancing the requirements for shortened total vehicle development time with longer development times for software, more stringent safety requirements, and, especially, growing complexity due to the rising number of functions and the increasing interaction between them. To master these challenges, automakers and suppliers are driving a paradigm change in software development from hand-coded to model-based development. The model-based development process is specifically attractive for automotive software as development in this domain is driven by two strong forces. On one side is the evolutionary development of automotive control systems, dealing with the iterated integration of new functions into a substantial amount of existing/legacy functionality from previous system versions, and on the other side the platform-independent development that substantially reduces the amount of reengineering/maintenance caused by fast-changing hardware generation. As a result, using a model-based approach in the development process in its early phases is being pursued to enable a shift in focus from function-based to code-based engineering of automotive systems. Model-based development is used by several automakers and suppliers even though no major empirical investigations into costs and benefits of model-based development have been conducted. Thus, the criteria for optimizing the costs and benefits of model-based development have to be identified first to determine the potential for further model-based development in development phases such as requirements engineering and E/E architecture design.

4.5.3 Hardware-in-the-Loop Tests

The functional and spatial distribution of ECUs (see Sect. 4.2.5) in today's vehicles has resulted in multifaceted innovations, such as DAS (see Sect. 4.2.4), ADAS (see Sect. 4.9 and Chap. 11), and other essential vehicle systems. A variety of tests ensures the reliability of these complex, real-time, networked vehicle systems to prevent potential malfunctions. With regard to complexity and real-time behavior, hardware-in-the-loop (HIL) test systems are used in the development and test of these complex real-time vehicle systems and have been established as a method for quality assurance of vehicle systems or components. HIL was introduced as a measure to improve the test possibilities in the automotive sector. Therefore, it can be seen as a method for testing and securing vehicle ECUs or mechatronic components and systems during development and early commissioning, providing an effective platform for the complexity of the vehicle ECUs or mechatronic components under test by adding a mathematical representation of the respective dynamic systems. The mathematical representation is referred to as the vehicle system simulation. In the automotive domain, HIL is applied in two main forms for the test:

- Adaptation of an ECU to a HIL simulator as a so-called component or module test bed
- Adaptation of several ECUs to one or several coupled HIL simulators as a so-called integration test bed

When performing HIL tests, the manually conducted tests in the initial phase are replaced by automatic test sequences performed with HIL test equipment (see Sect. 4.5.3.1). HIL includes electrical emulation of real hardware components from the vehicle for testing in a simulated sensor/actuator environment. These electrical emulations act as an interface between the real vehicle system and the vehicle system model representation under test. Hence, the scope of HIL tests is primarily aimed at establishing logically functional errors of the control software. Therefore, the value of each electrically emulated vehicle component is controlled by the system simulation and is read by the embedded system model under test. The overall test range at the HIL can be roughly divided into three categories:

- *Component Tests*: Deals with the function of a single ECU. These tests primarily show the functional specification of an ECU.
- *Integration Tests*: ECUs are tested in the network. The test focus is primarily on communication between the ECUs. Global functions are tested, the sub-functions of which are implemented and distributed over several ECUs.
- *Diagnostic Tests of Functions Implemented in the ECUs*: By generating faulty or implausible conditions, it is possible to test whether the ECU or the ECU interconnector detects these states and responds to them in accordance with the design specifications.

Apart from the pure connection of ECUs to a HIL environment, there is also a variant for the mechatronic components available. In this case, a part of the mechanics is integrated into the control loop. This approach is often used in conjunction with electronic steering systems, whereby a part of the steering rod is coupled as a real mechanic to the HIL environment.

In contrast to the test in the vehicle, the ECUs in the HIL test facility are fully embedded in a virtually simulated environment. Almost all parameters of this simulation environment can be changed, meaning that the test situation in which the vehicle is currently located does not depend on the actual external conditions, as in a driving test, but can be specified. Moreover, the vehicle characteristics are part of the simulation environment so they can be determined by changing the virtual vehicle parameters, such as weight, motorization, and other essential features, which has an influence on the HIL test results which can simply be checked. In case the HIL test system also permits a change in the ECU device coding, which depends on the chosen HIL test system, a set of different equipment variants can be tested with the same HIL test setup without great effort.

As an example of a HIL test, the platform for the development of a vehicle antilock braking system (see Sect. 4.2.2) has embedded mathematical representations for each of the following subsystems for the system simulation:

- Dynamics of the brake system's hydraulic components
- Road characteristics
- Vehicle dynamics, such as suspension, wheels, tires, roll, pitch, and yaw

The value of each electrically emulated vehicle component is controlled by the system simulation and is read by the embedded system model under test.

4.5.3.1 HIL Test System Architectures

As described in a white paper by National Instruments (NI) (URL19 2017), a HIL test system consists of three primary components:

- *Real-Time Processor:* The core of the HIL test system. It provides deterministic execution of most of the HIL test system components, such as hardware I/O communication, data logging, stimulus generation, and model execution. A real-time system is typically necessary to provide an accurate simulation of the parts of the system that are not physically present as part of the test.
- *I/O Interfaces:* Analog, digital, and bus signals that interact with the unit under test. They are used to produce stimulus signals, acquire data for logging and analysis, and provide the sensor/actuator interactions between the ECU being tested and the virtual environment being simulated by the model.
- *Operator Interface:* Communicates with the real-time processor to provide test commands and visualization. Often, this component also provides configuration management, test automation, analysis, and reporting tasks.

Many HIL test systems use hardware fault insertion to create signal faults between the ECU and the rest of the system to test, characterize, or validate the behavior of the device under these conditions. To accomplish this, fault insertion units (FIUs) inserted between the I/O interfaces and the ECU, as shown in Fig. 4.16, allow the HIL test system to switch the interface signals between normal operation and fault conditions, such as a short-to-ground or open-circuit constraint.

Some vehicle systems use multiple ECUs that are often networked together to function cohesively. Although each of these ECUs may initially be tested independently, a system's integration HIL test system, such as a full vehicle simulator, is often used to provide more complete virtual testing, as shown in Fig. 4.17.

Even with the latest multicore processing power, some vehicle systems require more processing power than what is available in a single HIL environment. To address this challenge, distributed processing techniques are used to meet the

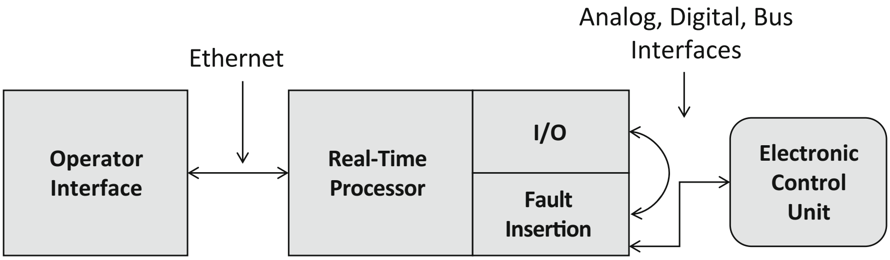


Fig. 4.16 Hardware fault insertion to test the behavior of the ECU during signal faults

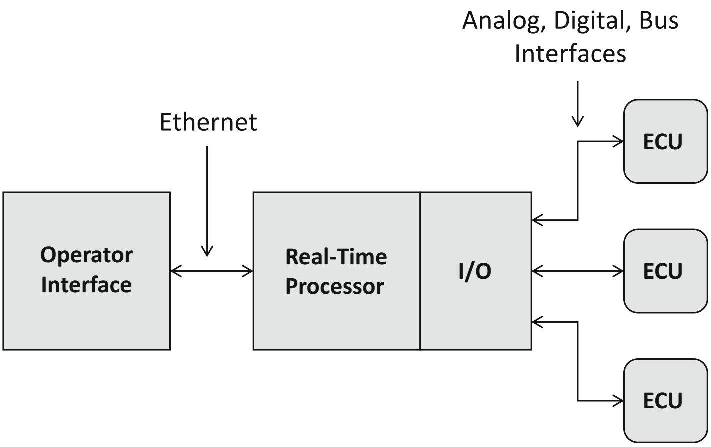


Fig. 4.17 Multiple ECU tests

performance requirements of these systems. In very high-channel-count systems, the need is more than simply additional processing power; additional I/O is also necessary. In contrast, systems using large, processor-hungry models often use additional real-time processors only for the extra processing power, allowing those processors to remain dedicated to a single task for greater efficiency. Depending on how the simulator tasks are distributed, it may be necessary to provide shared trigger and timing signals between the real-time processors as well as deterministic data mirroring to allow them to operate cohesively, as shown in Fig. 4.18.

Implementing and maintaining wiring for high-channel-count systems can pose costly and time-consuming challenges. These systems can require hundreds to thousands of signals be connected between the ECU and the HIL test system, often spanning many meters to compensate for space requirements.

Fortunately, deterministic distributed I/O technologies can help tame these wiring complexities and provide modular connectivity to ECUs, which allows for efficient system configuration modifications. Instead of routing all connections back to a single rack containing one or more real-time processing facilities instrumented with I/O interfaces, deterministic distributed I/O can be used to provide modular I/O interfaces located in close proximity to each ECU without sacrificing the high-speed determinism necessary for accurate simulation of the virtual parts of the system.

This approach greatly reduces the cost and complexity of HIL test system wiring by making it possible for the connections between the ECU and the I/O interfaces to be made locally (spanning less than a meter), while a single bus cable is used to span

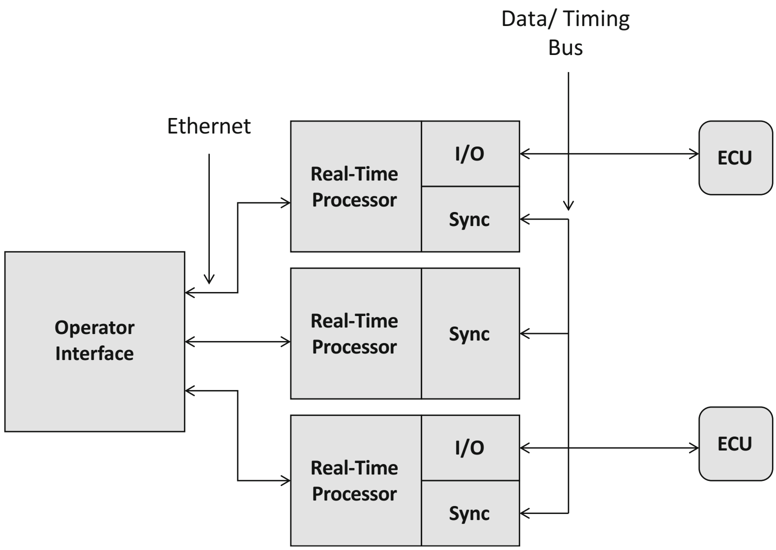


Fig. 4.18 Multiple real-time processors for additional processing power

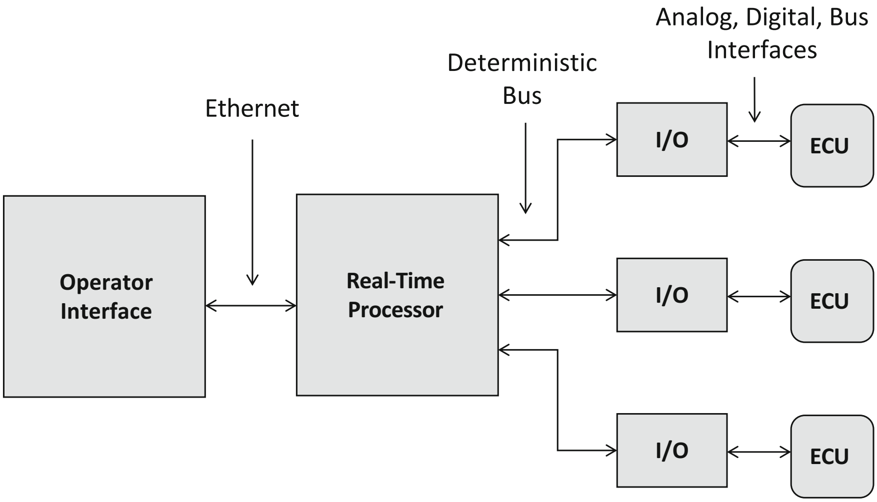


Fig. 4.19 Deterministic distributed I/O interfaces reduce HIL test system wiring cost and complexity because connections between ECU and I/O interfaces can be made locally

the additional distance to the real-time processing chassis. Additionally, with the modular nature of this approach, HIL test systems can easily scale incrementally, from a multi-ECU test system, in which all but one of the ECUs are simulated, to a complete systems integration HIL test system where none of the ECUs are simulated. The architecture behind this approach is shown in Fig. 4.19.

Once the appropriate architecture for the HIL test system is selected, the first step in creating a HIL test system is to select the components that best meet the vehicle development requirements. National Instruments (NI) provides a wide variety of real-time processing and I/O options for implementing HIL test systems. Because they are all based on open industry standards, users are assured that NI always delivers the latest advances in PC technology for the HIL test system and always meets future test system requirements. The NI HIL platform is open and extensible, which means that it can adapt to changing system requirements. Because of its modular architecture, the NI HIL platform can be easily upgraded with additional functionality, which helps in future proof test systems and meets the requirements of the most demanding embedded software testing applications. In addition to the widest range of I/O on the market, NI offers software tools that

- Automate HIL tests
- Perform post-processing and report generation
- Map test results to requirements.

with regard to

- Real-time Processing
- Analog/Digital I/O
- Fault Insertion
- Bus Interfaces
- Instrument Grade and RF I/O Vision/Motion.

These tools help to perform a wider range of tests earlier in the software development process, which reduces overall development cost while improving product quality (URL19 2017).

4.5.3.2 HIL Test System

dSPACE offers a wide range of HILs, HIL-specific hardware, and related software for ECU testing, such as HIL Simulator Full-Size and HIL Simulator Mid-Size, offering an easy way to update existing HIL systems. The Simulator Full-Size is a very versatile HIL, offering a comprehensive range of adaption and configuration possibilities to meet customer-specific requirements. It consists of one or multiple racks, is up to 41 height units tall, and uses standard processor and I/O cards, making adaptions easy. dSPACE Simulator Full-Size can be used for any application up to simulating a complete virtual vehicle and is shown in Fig. 4.20.

Fig. 4.20 dSPACE Full-Size HIL

Typical fields of application of dSPACE Full-Size HIL:

- Battery management systems
- Comprehensive closed-loop tests on ECUs, release/acceptance tests
- Electric motor simulation for hybrid or electric power trains and electric steering system
- Engine, power train, chassis, and body
- Mechanical test benches
- Networked ECUs
- Racing applications (Formula One, rally)
- Special requirements, e.g., with high system flexibility or high-current applications
- Truck applications

dSPACE Simulator Mid-Size generates and measures I/O signals via integrated dSPACE I/O boards. The function range is complemented by load and failure simulation. Typical fields of application of dSPACE Simulator Mid-Size HIL are:

- Automated testing
- Electric drives applications (combined with DS5202 Electric Motor HIL Solution or DS5203 FPGA Board)
- Engine, transmission, vehicle dynamics, and body electronics HIL
- Function integration tests, release tests, and ECU diagnostics tests
- Open-loop or closed-loop environment
- Realistic unit tests
- Real-time simulation

A standard dSPACE HIL Simulator Mid-Size also supports electrical failure simulation on all ECU output pins connected to the HIL I/O board. A hardware extension allows electrical failures to be simulated on ECU inputs as well. The host PC controls both types of failure simulation via an RS232 interface:

- Broken wire simulation (open circuit)
- ECU inputs optional by DS793/DS794 FIUs
- ECU outputs per load/FIUs
- Remote-controlled with ControlDesk® failure simulation and automated with AutomationDesk
- Simulation of cross-wired short circuits between ECU pins via common fail planes
- Simulation of short circuits: from ECU pins to ground or battery voltages
- Simultaneous activation of multiple failures (latch mode)

Another HIL simulation environment is SCALEXIO®, the main dSPACE HIL simulator ranging from small to large systems and providing very high processing power. It is configured entirely by software, which makes adapting to changing requirements easy and simple. SCALEXIO can also be coupled with other dSPACE HIL systems, such as the HIL Simulator Full-Size and the HIL Simulator Mid-Size, offering an easy way to update existing HIL systems.

dSPACE SCALEXIO is a very versatile technology that provides highly flexible channels that can be extended to any required size and is completely software-configurable. Its application range covers all test domains, including the test of ECUs of electric drives. With regard to the SCALEXIO multiprocessing feature, the simulator can be coupled with existing SCALEXIO-based or DS100x-processor-board-based systems, allowing users to expand their existing test setups to meet growing project needs. The key benefits are:

- Easily resizable to fit specific test tasks because component test systems and network systems are both built with the same standardized hardware components and connections
- Graphical configuration of channels
- Support of different workflows and user roles by separating I/O configuration, modeling, and code generation
- Support of functional mock-up interface (FMI)
- Test of different ECU variants and types on a single system with minimal configuration effort
- Use of virtual ECUs (V-ECUs) for HIL tests if the real ECU prototype is not available yet

Furthermore, SCALEXIO contains an FIU consisting of several components:

- An onboard failure routing unit (FRU) on the I/O channels prepares failure simulation by switching the I/O channels to fail rails. The FRU is available for each channel on the MultiCompact and HighFlex boards and uses relays to provide the features of the central failure simulation unit to each channel.
- Depending on their properties, the channels are connected to the failure simulation system by the high-current (up to 80 A) or the low-capacitance (up to 1 A) fail rail. The low-capacitance fail rail for an optimized signal quality connects signal generation channels and bus channels to the central FIU. The high-current fail rail connects signal measurement channels to the central FIU.
- The central FIU is located on either the DS2642 FIU and Power Switch Board or the DS2680 I/O unit. The central FIU uses semiconductor switches for switching the failures. It switches very fast (pulsed switching), which makes it possible to simulate loose contacts or insert faults for a very precise duration.
- The fail rail segment switch is used to switch selected segments into the fail rails for failure simulation. This way, the conducting capacity can be minimized to avoid signal corruption, even for large simulation systems that have a high number of inputs/outputs or that are distributed across several cabinets.

The SCALEXIO FIU concept with an example of HighFlex I/O boards is shown in Fig. 4.21. The available failure types of the SCALEXIO FIU are listed in Table 4.5.

The dSPACE tool chain also allows rapid prototyping and ECU validation with virtual test drives covering the following applications:

- Rapid control prototyping
  – Predictive drivetrain control for commercial vehicles
  – Autonomous emergency braking based on radar and camera data
  – Automatic windshield wiper control and rain sensor

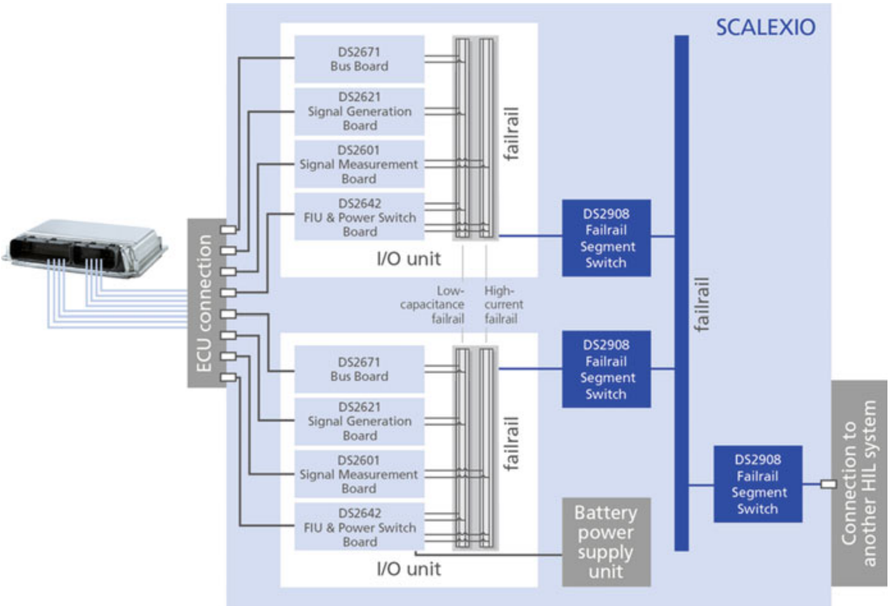


Fig. 4.21 SCALEXIO FIU concept with a selection of HighFlex I/O boards

Table 4.5 SCALEXIO failure types

Failure type	Failure on single signals	Failure on several signals
Open circuit	1 channel	All channels[a]
Short circuit to ground or U_{BAT}	2 channels	Up to 10 channels[a,b]
Short between channels	2 channels	Up to 10 channels[a,b]
Failure with pulsed switching	√	

[a]Requires the option “Activation by FRU relay” and is only possible on I/O channels without current enhancement

[b]Depending on the ampacity of the fail rail

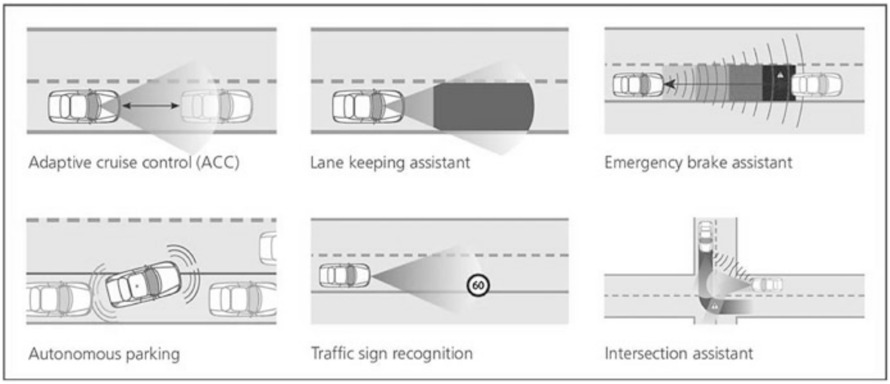


Fig. 4.22 dSPACE practical examples for virtual validation and hardware-in-the-loop simulation

- Virtual validation and hardware-in-the-loop simulation
  - Adaptive cruise control
  - Lane keeping assistants
  - Pedestrian detection
  - Traffic sign recognition
  - Intersection/cross-traffic assistant

These use case examples are illustrated in Fig. 4.22 used with the permission of dSPACE (URL20 2017).

4.6 AUTOSAR

The electric/electronic (E/E) architecture landscape in the automotive industry was characterized by proprietary solutions in the past, which seldom allowed the exchange of applications between automakers, and Tier 1 suppliers. With regard to the continued exponential growth in complexity and functionality of E/E components and systems, further proliferation of proprietary solutions consumes more and more resources and becomes difficult to control. This has resulted in an industry-wide initiative to manage the complexity of emerging automotive E/E architectures, the so-called AUTomotive Open System ARchitecture (AUTOSAR). AUTOSAR is a worldwide joint initiative of several major industries which was formed in mid-2003 to create and establish standardized software architecture for automotive ECUs. It serves as a platform upon which future vehicle applications can be implemented and also serves to minimize barriers between functional domains. Development goals include scalability to different vehicle and platform variants, transferability of software, consideration of availability and safety requirements, collaboration between various partners, sustainable utilization of natural resources, maintainability throughout the whole product life cycle, and process management during the entire life cycle of a product from inception, through engineering design and manufacturing, to service and disposal of manufactured products. AUTOSAR is driven by the advent of innovative vehicle applications, contemporary automotive E/E architecture that has reached a level of complexity requiring a technological breakthrough in order to manage it satisfactorily and fulfill the heightened passenger and legal requirements. This need is important for vehicle manufacturers and their leading Tier 1 suppliers who are faced with often conflicting requirements from:

- *Driver Assistance and Dynamic Drive Aspects*: Key items include detection and suppression of critical dynamic vehicle states and navigation in high-density traffic surroundings.
- *Legal Enforcement*: Key items include environmental aspects and safety requirements.
- *Passenger Convenience and Service Requirements:* Comfort and entertainment functional domains.

Leading OEMs and Tier 1 suppliers, having recognized this industry-wide challenge, decided to work together to meet the challenge. Their common objective is to create a development base for industry collaboration on basic functions while providing a platform which continues to encourage competition on innovative functions. To this end, a development partnership called AUTOSAR was formed, including all vehicle domains with the goals of (URL21 2017):

- Collaboration between various partners
- Definition of an open architecture
- Development of highly dependable systems
- Scalability to different vehicle and platform variants
- Standardization of basic software functionality of automotive ECUs
- Support of different functional domains
- Support of applicable automotive international standards and state-of-the-art technologies
- Transferability of software

The AUTOSAR standard serves as a platform upon which future vehicle applications will be embedded and also serves to minimize the current barriers between functional domains. It will, therefore, be possible to map functions and functional networks to different control nodes in the system, almost independently from the associated hardware. The technical goals of AUTOSAR:

- Modularity of automotive software elements to enable tailoring of software according to the individual requirements of ECUs and their tasks.
- Reusability of functions to help improve product quality and reliability and to reinforce corporate brand image across product lines.

- Scalability of function to ensure the adaptability of common software modules to different vehicle platforms and prohibit proliferation of software with similar functionality.
- Transferability of functions to optimize the use of resources available throughout a vehicle's electronic architecture.

This helps to provide a common software infrastructure for automotive systems of all vehicle domains based on standardized interfaces for the different layers, as shown in Fig. 4.23. This common infrastructure encompasses the following elements:

- *Electronic Control Unit (ECU)*: The physical hardware.
- *Runtime Environment (RTE)*: All communication between software components and basic software including the operating systems (OS) and communication services is carried out through the RTE layer.
- *Main Software*: A combination of:
  - *Basic Software:* Builds on RTE to offer some general utilities which provide the overall functionality of the AUTOSAR infrastructure (software components and RTE on an ECU). Basic software is essential for running the functional part of the software; however, it does not fulfill any functional job itself. The software components do that.

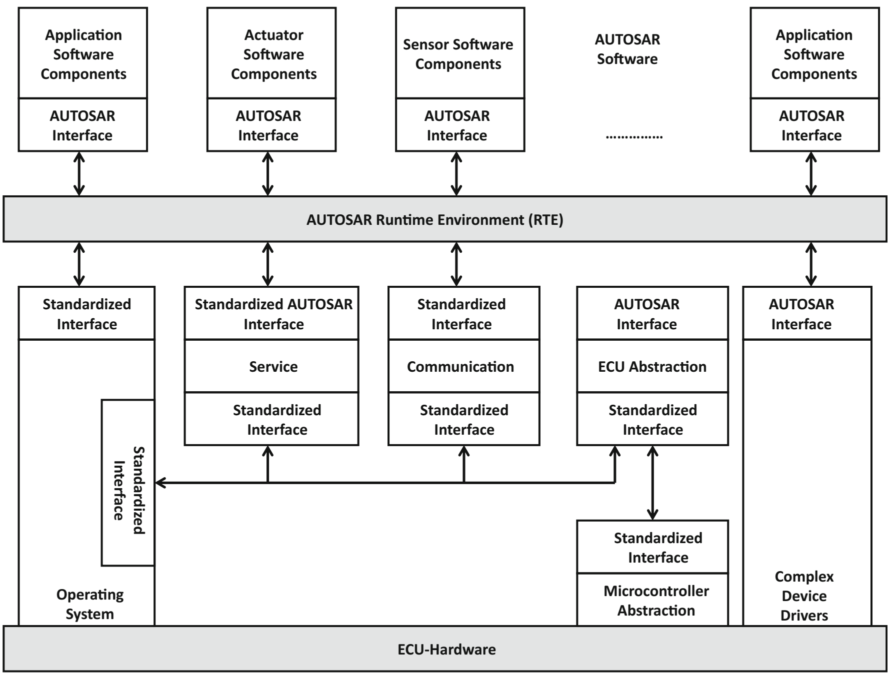


Fig. 4.23 AUTOSAR ECU software architecture (modified after (URL21 2017))

  - *Software Components:* Building blocks for software systems, either custom made or purchased off-the-shelf, each supporting and implementing a dedicated set of functionalities, and, in conjunction, providing the overall functionality of the software application. Software components are the fundamental building blocks of AUTOSAR systems. Types of software components are:
    - Application software components
    - Actuators/sensor software components
- *Complementary Software*: Manufacturer- and model-specific software.

The AUTOSAR RTE is the central connecting element in an AUTOSAR ECU architecture. It realizes the interfaces to enable interaction between any kind of AUTOSAR software components. Each ECU component has its own customized RTE implementation. Depending on the location of each component, the former virtual interaction can then be mapped to a real interaction implementation. Components that are mapped onto one ECU communicate through intra-ECU mechanisms. Since the RTE source code is generated, it can be tailored by the generator to implement the communication paths required by its connected AUTOSAR components. Thus the RTE can be interpreted as a static implementation of specialized communication topologies.

The ECU abstraction layer provides a unified interface for AUTOSAR software components to access electrical values of the underlying ECU independently of the actual ECU hardware architecture. The ECU abstraction itself is closely coupled to the microcontroller abstraction layer that provides access to the actual physical signals of the microcontroller. The microcontroller abstraction layer is a hardware-specific component available on each standard microcontroller and provides the basic software to access hardware information without directly accessing the microcontroller's registers. Among others, MCAL provides access to digital I/O, analog/digital converter, flash, electrically erasable programmable read-only memory (EEPROM), and others.

Hence, standardization of functional interfaces across automakers and suppliers and standardization of the interfaces between the different software layers is seen as a basis for achieving the technical goals of AUTOSAR. AUTOSAR provides a standard description format for the interfaces as well as other aspects needed for the integration of the AUTOSAR software components. Key ECU automotive software elements are:

- *Operating System*: Task scheduler (event, take place on a regular basis, etc.)
- *Application*: Supports normal power train operations; diagnostic, calibration
- *Network*: Communication, data transfer, OEM network strategy (Ford Network Operating System, General Motor Local Area Network, etc.), and data transfer

Some principal classical challenges and solutions suggested by AUTOSAR, together with their implied benefits, are listed in Table 4.6 (Heinecke et al. 2003).

Table 4.6 Challenges, solutions, benefits of AUTOSAR

Challenges	Solutions	Benefits
Non-competitive functions have to be adapted to OEM-specific environments	Standardized interfaces	Reduction/avoidance of interface proliferation within and across OEMs and suppliers
Tiny little innovations cannot be implemented at reasonable effort as provision of interfaces from other components requires a lot of effort		Eases implementation of hardware-independent software functionality by using generic interface catalogs
Missing clear interfaces between basic software and code generated from models		Simplifies model-based development and makes it usable for standardized AUTOSAR code generation tools
		Reusability of modules across OEM
		Exchangeability of components from different suppliers
Effort wasted on layout and optimization of components which add no value recognized by customer	Basis software core	Enhanced software quality
		Concentration on functions with competitive value
Obsolescence of hardware (μC, circuits,. . .) causes huge efforts in adapting existing software	Microcontroller abstraction	Part of the hardware can be exchanged without need for adaptation of higher software/ functions/applications
Extended needs for microcontroller performance (caused by new functions) causes need for upgrade, i.e., redesign effort		
Large effort when relocating functions between ECUs	Runtime environment	Encapsulation of functions creates independence of communication technology
Large effort when reusing functions		Communication easier through standardized mechanisms
		Partitioning and relocatability of functions possible
Immature processes because of acting in ad hoc mode/missing traceability of functional requirements	Software component template	Improvement in specification (format and content)
Lack of compatible tooling (supplier, OEM)	Exchange formats	Opportunity for a seamless tool chain
OEM buys black box and is not able to extend/integrate new functionality in an ECU (e.g., integration of tire guard functionality)	Technical integration of software of multiple suppliers	Eased process of integration of different software components allows optimization of hardware costs
Lack of guidelines for use/buy of software components	Conformance test process	Integration of third-party software components
Unclear legal situation	License agreement	Common understanding between suppliers and OEMs

4.7 AUTOSAR Adaptive Platform

In the near future, domain controllers will be enhanced with multicore processors in vehicles used for computing-intensive vehicle applications. Moreover, the vision for autonomous driving also requires such domain controllers which results in more sophisticated designs. Adaptive AUTOSAR, the next generation of AUTOSAR, is software on which such designs can be based.

The AUTOSAR Adaptive Platform is designed to support software engineers creating more flexible E/E architectures. For this reason, the AUTOSAR Adaptive Platform will provide a software framework for more complex vehicle systems. Engineers will be supported by an increase in bandwidth, the result of implementing Ethernet networking technology "to provide an optimal standardized software framework for new applications, especially in the fields of connectivity and highly automated and autonomous driving," said Stefan Rathgeber, spokesperson for the AUTOSAR development partnership (URL22 2017). Thus, classic and adaptive applications can be seamlessly combined using an Ethernet connection.

Therefore, the new standard will probably be first applied in an ADAS. However, "highly automated driving systems must be dependable and have fail-safe operational capabilities," Rathgeber explained. This can only be accomplished with features such as high data processing capacities, service-oriented communication, and over-the-air updates. The driverless vehicle is where the new platform's strengths will ultimately provide the greatest benefits, being a key enabler on the way to a self-driving vehicle by making the new platform accessible to as many manufacturers, suppliers, and developers as possible. It can also aid infotainment system development by providing a more seamless integration into a standard operating system with more connectivity and graphics computing power.

Among the development committee's goals is to create a dynamic system that includes middleware and supports complex operating systems using a POSIX interface and multicore microprocessors. Its main communication approach is based on service-oriented communication and IP/Ethernet. The platform will be capable of supporting adaptive software deployment while interacting with non-AUTOSAR systems, as shown in Fig. 4.24.

4.8 GENIVI

Compared to AUTOSAR, the non-profit GENIVI Alliance is committed to driving the broad adoption of specified, open source, IVI software. Therefore, GENIVI provides automakers with four unique approaches to meeting today's challenges:

1. *Define:* Allows flexible definition of IVI systems that fit customers' latest needs
2. *Partner:* Supports business model evolution and networking across the supply chain
3. *Leverage:* Provides standard, open source architectures, tools, and software components
4. *Reuse:* Allows reuse of components and redeployment of solutions with no royalty fees

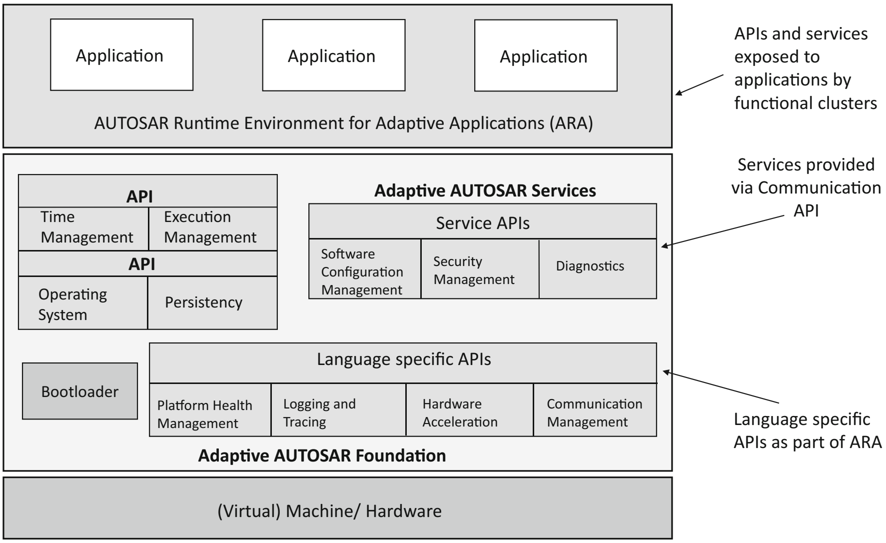


Fig. 4.24 Adaptive AUTOSAR platform

Automakers and their suppliers face at least three significant challenges in developing and delivering IVI functionality to their customers:

- *Responding to Consumers:* Consumers want IVI functionality that is the same or similar to that found in consumer electronic devices, such as smartphones and tablets. New devices with the latest features are typically launched in the market on an 8- to 18-month cycle versus the 2–5 year cycle for most in-vehicle software. As a result, consumers have introduced a new competitive measure that automakers must use: the time from consumer request to in-vehicle availability.
  - GENIVI's open software approach better aligns consumer electronics and automotive development cycles.
  - GENIVI's individual software components and reusable platform provide automakers and their suppliers with the tools to perform rapid prototyping and to quickly develop and deliver IVI systems that fulfill consumer requests.
- *Complexity and Cost:* Consumer functionality requests push the amount of software in a typical IVI system to over several million lines of code. Hence, automakers have to deal with the increasing complexity and cost of developing, validating, and maintaining software. Many automakers are shifting away from the historic black box approach and are taking more ownership of the design and development process, including maximizing the reuse of legacy code to reduce costs and deploying a software platform on multiple hardware platforms based on the needs of their various models.
  - GENIVI's technical deliverables and open approach promotes a wide range of supplier models based on the preferences of the automaker.

  - Automakers can launch a single reusable software platform that, with limited integration, can run on a wide range of automotive boards, from low- to high-end performance.
- *Customer Ownership:* Automakers are keen to keep their customer relationships sustainable. Large technology companies, such as Apple and Google, have entered the automotive market, introducing demands for user experience, branding, and data usage that limit the automaker-driver relationship. Automakers have their own business model; some prefer a single Tier 1 supplier, while others prefer multiple suppliers taking ownership of certain pieces of the overall system.
  - GENIVI's approach allows automakers to maintain their independence from technology titans by pushing their own business models in the automotive industry.
  - GENIVI's flexible architecture and pick-and-mix model gives automakers the freedom to include preferred, best-in-class software from multiple suppliers.

GENIVI's technical deliverables consist of:

- Flexible technical architecture
- Individual software components
- Pre-integrated, reusable IVI platform
- Standard interfaces/application programming interfaces (APIs)

that are essential to overcoming the IVI challenges faced by every automaker. Thus, GENIVI technologies is at the forefront of a new generation of IVI solutions. As one of the many GENIVI use cases, BMW has moved from its traditional approach to IVI software development to where it is today, the first automaker to deliver a complete infotainment product, the so-called entry media and navigation system (EMNS). EMNS rolled off the assembly line in the fall of 2013 and is now in the MINI and 1, 3, and 5 BMW series product lines based on the GENIVI Linux platform. Since then, other automakers have selected products with GENIVI solutions resulting in vehicles on the road in four continents around the world. Furthermore, several additional automakers will release GENIVI-equipped systems in vehicles during the next 2 years.

4.9 Example: Advanced Driver Assistance System

Advanced driver assistance systems (ADAS) (see also Chap. 11) support vehicle drivers in the driving process by enhancing it for safety and better driving conditions. Therefore, safety features are implemented to avoid collisions or accidents by embedding intelligent safeguard devices and taking over control of the vehicle in critical driving or traffic situations. In this regard, the ADAS development process began with the definition and specification of functional requirements in terms of the desired safety functions being embedded ride comfort, and operational restrictions. The primary functionality of ADAS is to facilitate the task performance of drivers by providing them with:

- Instructions
- Real-time advice
- Warnings

This type of driver support operates in different kinds of modes (Rosengren 1995), such as:

- Advisory mode
- Automatic mode
- Semiautomatic mode.

All of them have different consequences for the driving task and hence on vehicle and traffic safety. Thus, the purpose of ADAS is to reduce or even eliminate driver errors, resulting in an enhanced efficiency in driving the vehicle. Therefore, the benefits of ADAS are high because of a significant decrease in human suffering, economical costs, and less pollution because:

- Drivers using ADAS will be safe and efficient drivers.
- Driving safety will be considerably enhanced.
- High-performance driving is possible without regard to vision, weather, and environmental constraints.
- More vehicles will be accommodated on regular highways but especially in dedicated lanes.
- Other essential features of safe driving.

In this regard ADASs are safety-critical systems that require a high degree of:

- *Fault Tolerance*: Enables a system to continue its intended operation in the event of a failure of one or some of its components, possibly at a reduced level, rather than failing completely.
- *Real-Time Behavior*: A real-time system that executes tasks to completion in a guaranteed amount of time.
- *Reliability:* The ability of a system or component to perform its required functions under stated conditions for a specified time.
- *Security:* The degree of resistance to, or protection from, harm.

Functional and safety requirements can be represented by the system specification in order to define the exact operation of safe system functionality. Therefore, the system specification represents the basis for the top-level design of the system architecture, followed by a detailed module design of:

- Actuators
- Controller
- Driver interfaces
- Human-machine interface (HMI)
- Sensors
- Other essential components

After implementation of the various hardware and software modules, the system will be carried out by integrating the individual modules, bringing overall system functionality and safety together. In each integration step, verification is performed to determine whether or not the output of a step meets the design specifications because ADAS relies on inputs from multiple data sources. However, additional inputs are possible from other sources separate from the primary vehicle platform, such as other vehicles, referred to as V2V or V2I, and the vehicle-to-X (V2X) systems.

More in general, ADAS (see Chap. 11) is one of the fastest-growing segments of automotive electronics with industry-wide standards in vehicular safety systems, such as ISO 26262, and the developing technology specific standards, such as IEEE 2020 for image sensor quality, and communication protocols, such as the vehicle information API, as reported in (URL23 2017). In general, an API is a set of routines, protocols, and tools for building software-based applications, specifying how software application programs should interact.

ADAS design can be done through so-called model-in-the-loop (MIL)testing and simulations to abstract the ADAS behavior in a way that the developed ADAS model can be used for testing, simulating, and verifying. Using an industry standard, such as Simulink, for model definition enables the engineer to test and refine the model within a desktop environment, allowing a complex system to be developed efficiently (URL24 2017; URL25 2017). The code can subsequently be used with software-in-the-loop (SIL) simulations verified by the remaining hardware components, vehicle dynamics, and simulation of the real-time environment. Finally the hardware can then be tested by a real-time (HIL) simulation (see Sect. 4.5.3).

4.9.1 ADAS Functionalities

With regard to:

- Advanced functions
- HMI
- Information distributions
- Sensors and actuators
- Systems software and hardware

ADAS can cover a variety of functionalities, and many similar systems in different parts of the world often exist with a slightly different name (see Chap. 11). The systems functionalities mentioned on the website of the ADASE2 cluster (URL26 2017) and some additional ones are listed alphabetically below:

- *ACC/Stop and Go and Foresight (ACC/SaG)*: Ensures that during a stop and go traffic situation, the longitudinal control of a vehicle will be partly carried out by detecting the traffic in front of the vehicle in the near field. In extension to an ACC system, the detection of the near-field area is necessary to react to other vehicles swerving into the near field as they react to the more far away traffic situation of the vehicles ahead. Hence, the near-field communication at the end of a traffic jam can be included into the longitudinal control of a vehicle before the driver is able to see it.
- *Automatic Parking (AP)*: Ensures that a driver entering into a parking slot in a parallel maneuver is supported by automatically taking over the steering and engine control
- *Autonomous Driving (AD):* Ensures that driving is safely controlled in every situation by an algorithm.
- *Autonomous Emergency Breaking (AEB)*: System that avoids a collision with another vehicle or a pedestrian.
- *Blind Spot Detection (BSD)*: Sensors monitor the road area behind and next to the vehicle and warn if the driver tries to pull out despite there being no gap, taking much of the strain off the driver and avoiding hazardous situations.
- *Intelligent Headlight Control (IHC)*: Ensures optimum illumination of the road. The improved vision makes driving at night much safer and more comfortable. The system uses a video camera to measure the ambient brightness and to estimate the distance from vehicles in front and oncoming traffic. This data is used to implement a variety of light functions. The high beam activation function enables drivers to use their high beam lights as often as possible without having to manually switch them on and off. If the function does not detect any other vehicles, it activates the high beam lights. However, if a vehicle is detected, the high beam light is switched off again. The adaptive high beam control function enables variable adjustment of the high beam range between the low beam and high beam levels. The area between the vehicle in front and an oncoming vehicle is better illuminated since the headlight cone is continuously adapted to this distance. With the continuous high beam control function, the driver can travel with permanent high beam headlights. If the camera detects other vehicles, the headlights are tilted horizontally or vertically, independently if necessary. This produces cones of light in which other road users are blocked out. The light distribution from the high beam lights remains virtually unchanged, while the driver's visual range is increased considerably.
- *Intersection Support (IS)*: Ensures certain tasks are supported, such as approaching stop signs, traffic lights, cross-traffic, and other tasks which result in IS system complexity due to the manifold scenarios which have to be supported by the system making detection and interpretation very difficult.

- *Lane Change Assistant (LCA)*: Ensures the driver is warned before and during a dangerous lane change. Through an acoustic and visual warning system, as well as a haptic feedback from the steering wheel, the driver is supported following a lane change trajectory. This requires detection of all other vehicles around the vehicle as well as the detection of the lane. LCA consists of several combined systems, such as LDW and BSD. Functional limits of LCA systems are apparent in the case of fast approaching vehicles.
- *Lane Departure Warning (LDW):* This Continental-developed driver assistance system alerts the driver with acoustical or haptic warnings before the vehicle is about to leave the lane. According to a study carried out on behalf of the German Federal Ministry of Education and Research, LDW could prevent about half of the accidents caused in this way, as reported in (URL27 2017). In Fig. 4.25, the LDW system function is depicted, based on the example given in (URL27 2017).
- *Lane Keeping Assistant (LKA)*: Responds through a gentle intervention in the steering, which the driver can counteract at any time. This can save additional reaction time in cases where each and every second counts.
- *Local Hazard Warning (LHW)*: In a case where a hazard occurs too far in front of the vehicle for the driver to see it, the LHW system will warn the driver by communicating hazard information, e.g., an accident far down the road is passed along over long distances by using ad hoc networks.
- *Near-Field Collision Warning (NFCW)*: Detects vehicles in the near field, such as in the blind spot area (see BSD). The detection area is very close to the vehicle in the near field. Driver warning can be acoustical, haptical, or optical.
- *Night Vision Plus (NVP)*: Offers the driver a true-to-life image of the road ahead and provides valuable information about the course of the road, vulnerable road users, and obstacles on and alongside the road. When the system identifies pedestrians, they are highlighted clearly in the night vision image. Night vision plus directs the attention of the driver to potential risks, allowing the driver to take appropriate action.
- *Obstacle and Collision Warning (OCW)*: The driver will be warned if a potential collision is detected with, e.g., another vehicle or obstacle. The warning can be acoustic or visual. Complex scenarios, such as evading, can be included as well as warning breaking, which is a very short brake in order to give a kinesthetic feedback.

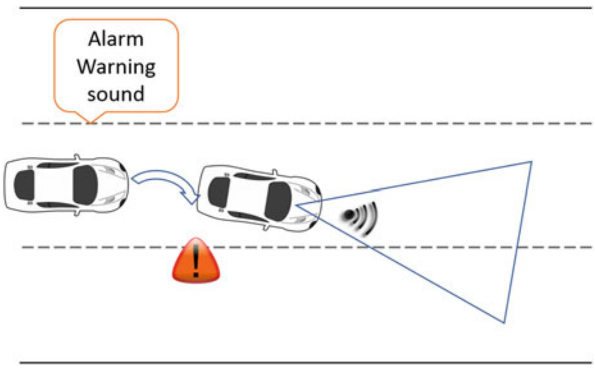


Fig. 4.25 LDW driver assistance system detecting vehicle's lane-following behavior (URL27 2017)

- *Platooning:* Ensures that several vehicles, e.g., trucks, that are following one after the other in a platoon in order to save space are connected electronically by means of communication.
- *Pre-crash Collision and Mitigation System (PCCMS):* Ensures damage reduction from an accident by acting on the pretensioner of the safety belts before the accident occurs and automatically starting to brake when the system detects an upcoming collision which cannot be avoided.
- *Rear View System (RVS):* Many new vehicle models allow drivers to see very little of the vehicle's immediate environment. As aerodynamics and pedestrian protection issues exert an ever greater influence on vehicle contours, while side and rear windows shrink in size, it is becoming almost impossible to maneuver cars safely and precisely. The Bosch rear view system supports drivers as they reverse their vehicle. The camera image is displayed via the radio or radio navigation system and shows the area behind the vehicle.
- *Road Departure Protection (RDP)*: The Continental RDP avoids roadway departure crashes, which currently are not completely covered by today's lateral guidance ADAS. The Continental base system uses a forward-looking mono camera to detect roadway boundaries, monitor the driver's steering angle and vehicle path through existing ESC sensors, and use chassis motion sensors to identify if the vehicle is crossing the road boundary. It then uses the existing ESC system to apply the individual wheel brakes to automatically steer the vehicle back on the road while simultaneously warning the driver and reducing the speed of the vehicle for safety reasons. This active intervention is signaled when the vehicle senses it is departing from the road. The system is designed with a driver intention recognition feature in the event that the driver does intend to leave the roadway for any reason (URL28 2017). In Fig. 4.26, the LDW system function is shown, based on the example given in (URL28 2017).
- *Road Sign Recognition (RSR):* Drivers should always be aware of the current speed limit. With road sign recognition, currently applicable road signs are in view. When a video camera identifies a road sign signaling the beginning or end of a speed limit as well as any special instructions, such as slippery when wet, the function displays this sign in the form of a symbol in the cockpit. The speed limit on variable message signs as well as restrictions on overtaking and the end of such restrictions can also be detected. If the driver fails to observe the speed limit, he

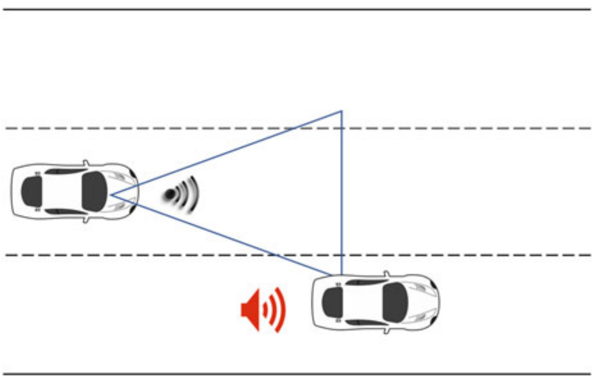


Fig. 4.26 RDP driver assistance system detecting vehicle's lane-following behavior (URL28 2017)

can, for example, be warned by an audible signal. In the future, it will also be possible to detect other road signs in addition to speed limits.

- *Rural Drive Assistance (RDA):* Ensures that systems developed for use on highways also work on rural roads which require the extension of some system functionalities of the highway systems.
- *Traffic Sign Recognition (TSR):* Has a display on the instrument panel to remind drivers of the current speed limit. This is achieved through multiple uses of the same camera which is also used for the lane departure warning system. When combined with high-performance software, it can also recognize speed limit signs. Digitized speed limit information of the onboard navigation system will be incorporated to prepare for roads without assigned speed limit signs.

It has to be mentioned that the previous alphabetic list of functionalities of ADAS is, in some cases, related to a research roadmap. Obstacle and collision avoidance, platooning, and autonomous driving are still being researched and will be developed for use in the near future in conjunction with the development of the required sensors for the aforementioned advanced functionalities. In addition, ADAS and pedestrian protection systems (PPSs) have become an active research area aimed at improving traffic safety. In this regard, the major challenge of PPSs is the development of reliable onboard pedestrian detection systems. Due to the varying appearance of pedestrians with regard to the following, it is very difficult to cope with the robustness needed for this kind of protection system (Gironimo et al. 2009).

- Aspect ratio
- Different sizes
- Different types of clothing
- Dynamic shape
- Unstructured environment

Thus, the problems arising in this research area are the lack of public benchmarks and the difficulty in reproducing many of the proposed methods, which makes it difficult to compare the approaches. Hence, a more convenient strategy for surveying the different approaches will be dividing the problem of detecting pedestrians from images into different processing steps, each with attached responsibilities. Then, the different proposed methods can be analyzed and classified with respect to each processing stage, favoring a comparative viewpoint.

4.9.2 ADAS Sensor Types

Sensors are designed for specific application domains in which they work over a specific range. The design range is usually determined with regard to the application, ensuring safe and precise measures. The reason is that if the measuring range is exceeded, the sensor may be permanently damaged or destroyed. More in general, a sensor is a device that generates a measurable signal in response to a stimulus received from the following:

- Components
- Objects
- Systems

The characteristics of a sensor can be classified as being either static or dynamic, which is important in high-fidelity mapping of output versus input signals. Static characteristics are those measured after all transient effects have stabilized to their final or steady state. In contrast, dynamic characteristics describe the sensor's transient properties.

ADAS require different types of sensors such as vision and range sensors to accurately determine situational assessment and action implementation. Common sensor technologies for ADASs, which are being increasingly integrated by OEMs and Tier 1 suppliers, are:

- Infrared camera
- Lidar
- Radar
- Ultrasonic
- Video

According to market analysts, the use of forward-looking cameras will go up from 30 million units in 2014 to nearly 100 million by 2019. However, range sensors, which are based on radar and, more recently, on LiDAR technologies, are projected to be much more available in vehicles within the next 2 years. Furthermore, ongoing technology and integration developments in many ADAS application areas are increasing the design alternatives. Thus, OEMs and Tier 1 suppliers need to continually evaluate their systems, deciding how and when to integrate the newest technologies and latest advancements into their designs. In order to be able to use the sensor signals correctly, the operation of the respective sensor and the nature of the signals they generate must be clearly understood. With regard to this knowledge, engineers must be able to use the right approach for data acquisition from the sensor.

Looking at the manifold ADASs (see Sect. 4.9.1), a lot of different types of sensors are present, such as traditional technologies like cameras, radar, and ultrasound, which display various limitations in the context of ADAS applications, from sensitivity to weather conditions to the ability of reliable detection of objects. LiDAR, an advanced sensor technology for ADAS, remains cost prohibitive even at high volumes and may lack the robustness required for automotive applications.

Existing fixed-beam LiDARs are more robust than their scanning counterparts; however, they also entail major limitations in terms of distance range. A comparison of the main automotive detection and ranging technologies is shown in Table 4.7 (URL29 2017). To gain the respective knowledge about ADAS sensor operation and the nature of the signals they generate, the most important ones are discussed in more detail in the following subsections.

Table 4.7 Comparison of main automotive detection and ranging technologies

<table>
<tr><th></th><th>Ultrasonic</th><th>Camera</th><th>Camera</th><th>Radar</th><th>LiDAR</th><th>Laser</th></tr>
<tr><td>Impact of lighting conditions</td><td>None</td><td colspan="2">High</td><td>None</td><td colspan="2">Low to medium</td></tr>
<tr><td>Impact of weather conditions</td><td colspan="3">High</td><td>Low</td><td colspan="2">Medium</td></tr>
<tr><td>Field of view type</td><td colspan="2">Short and wide</td><td colspan="2">Far and narrow</td><td>Short and wide</td><td>Far and wide</td></tr>
<tr><td>No moving part design</td><td colspan="5">Yes</td><td>No</td></tr>
<tr><td>Pedestrian detection</td><td>Limited</td><td colspan="2">Yes</td><td>Limited</td><td colspan="2">Yes</td></tr>
<tr><td>Stationary object detection</td><td colspan="3">Yes</td><td>Limited</td><td colspan="2">Yes</td></tr>
</table>

4.9.2.1 RADAR Sensor

Radar is the acronym for radio detection and ranging. A radar system operates in the ultrahigh frequency (UHF) or microwave part of the radio-frequency (RF) spectrum and is used to detect the position and/or movement of objects, such as vehicles. Radar waves are transmitted at defined intervals. The delay between the transmitted wave and the echo determines the radial position for each azimuth direction on the display used. The greater the echo delay from a particular object in space, the farther from the display center it appears. Radar systems in vehicles are responsible for the detection of potential collisions or hazardous situations. A positive detection can be used to warn/alert the driver or to intervene with the braking and other controls of the vehicle in order to prevent an accident. In its practical realization, a vehicle radar system contains one or more radar sensors to detect obstacles around the vehicle and their speeds relative to the vehicle. Based on the detection signals generated by the sensors, a processing unit determines the appropriate action needed to avoid the collision or to reduce the collateral damage.

Using the vehicle radar system as a decision-making unit, it can:

- Alert the driver about any potential danger
- Assist the driver in parking the vehicle
- Prevent collisions by intervening with the control of the vehicle in hazardous situations
- Take over partial control of the vehicle such as adaptive cruise control

The key performance parameters of a vehicle radar system are:

- Angular resolution
- Angular width of view
- Detection range
- Range precision
- Speed detection range
- Velocity precision

With regard to the aforementioned characteristics, vehicle radar systems can be divided into three subcategories:

- Short range
- Midrange
- Long range

The main feature of short-range radars is range accuracy, while for midrange and long-range radar systems, the key performance feature is detection range. Short-range and midrange radar systems (a range of tens of meters) enable several ADAS applications, such as BSD, PCCMS, and LDW. They can also be used for implementation of SaG applications in city traffic. Long-range radars (hundreds of meters) are typically used for ACC systems. These systems can provide enough accuracy and resolution for even relatively high speeds of ~ 120 mph (URL30 2017).

The typical radar applications in ADAS are shown in Fig. 4.27.

4.9.2.2 LiDAR Sensor

LiDAR is applicable in a broad range of locating, profiling, and ranging applications. A LiDAR system consists of a laser capable of transmitting light (pulsed or

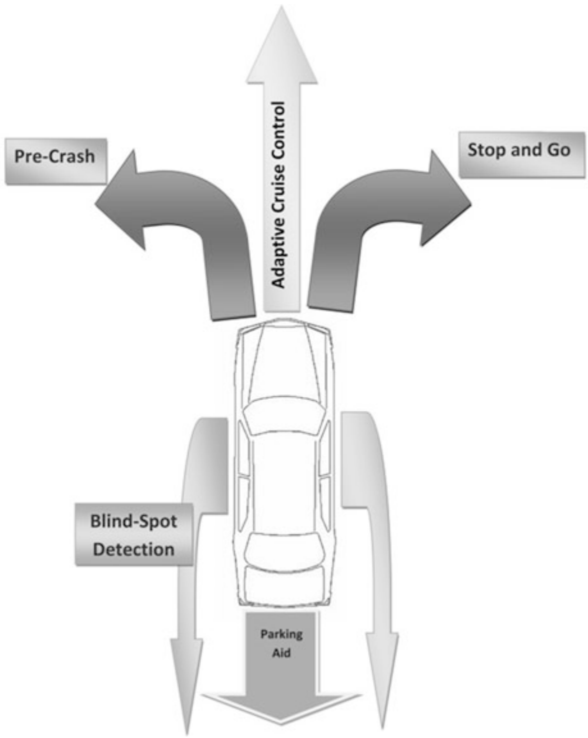


Fig. 4.27 Radar sensor applications in ADAS (URL30 2017)

continuous) over the required range of distance and a high-speed, low-noise receiver for reflected signal analysis. The transmitted light interacts with the target. A percentage of this light is reflected to the receiver according to the reflectivity of the target. Changes in the properties of the transmitted signal enable some properties of the target to be determined by providing clear-cut 3D snapshots of every object in the vehicle's vicinity. Along with its surveying feature, LiDAR has improved capabilities when it comes to the detection of objects, even in cases where there is a complete absence of light. LiDAR's features, which include ACC, BSD, and PCCMS and PPS, are not only better than the features of other sensors but are also far more consistent and reliable.

It is expected that LiDAR will become a central element of the autonomous vehicle's sensor suite, alongside existing technologies, ensuring robust sensing redundancy and increasing overall system reliability. The number of vendors in the automotive LiDAR sensor market is high, including companies such as Bosch, Continental, Denso, Hella, First Sensor, LeddarTech, Novariant, Phantom Intelligence, Quanergy, Teledyne Optech, Valeo, and Velodyne Lidar.

4.9.2.3 Laser Sensor

Laser light consists of light waves of the same wavelength which have a fixed phase relationship (coherence) resulting in an important feature of laser sensors: the almost parallel light beam. The resulting small divergence angle makes it possible to realize large ranges and topological mapping, both of which are required to build a sensor that provides high reliability for long distance measurements to ensure reliable vehicle guidance and measurements with regard to collision avoidance or the ability to narrow to a specific target. In this regard, blind spots can cause costly collisions. ADAS functions performed by laser sensor systems are LDW, LHW, NFCW, OCW, and PCCMS.

Weather conditions, such as fog or even dust, do not create a problem for laser sensors when they are embedded in a target discrimination mode. In this mode, the sensor performs a comparison and is able to distinguish the last target from all other reflections, which equates to dust or fog penetration and the ability of the sensor to see the road surface.

4.9.2.4 Camera Sensor

The front camera sensor is used in an ADAS machine vision system using images from a forward-facing camera to perform tasks such as lane-departure warning (LDW), obstacle and collision warning (OCW), traffic sign recognition (TSR), and distance measurement, among others. The output can either be a warning to the driver or direct control of certain vehicle functions, such as steering or braking.

The rear camera system uses images from a backward-facing camera to perform tasks such as automated parking (AP), object detection, and distance measurement, among others. The output can either be a warning to the driver or direct control of certain vehicle functions.

The surround-view-camera system is an ADAS technology that assists the driver in parking the vehicle safely by providing a top-down view of the 360° surroundings of the vehicle, as can be seen in Fig. 4.28 (URL31 2017).

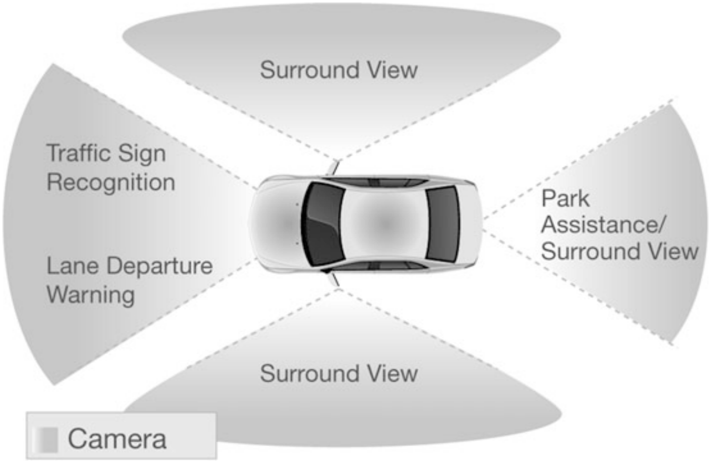


Fig. 4.28 Camera sensor applications in ADAS (URL31 2017)

4.9.2.5 Vision Sensor

Vision sensors use a camera to detect white lines and obstacles on the road ahead. A new sensor design by DENSO uses a pair of cameras placed side by side, which enable the distance to a target object to be measured more accurately and enhances the activation of autonomous emergency braking (AEB), lane-departure warning (LDW), and intelligent headlight system (IHS), a high beam system that automatically switches the headlights from high beam to low beam for better night vision.

Compact vehicles have limited space to install devices. Therefore, vision sensors are required to be small for easier installation. Generally, in stereo cameras, the longer the distance between the two camera lenses (baseline), the longer the measurable distance to the target, meaning that the camera body needs to be larger to extend the maximum measurable distance. A combination of highly accurate lens distortion correction and stereo matching technologies enable the new sensor to ensure that the maximum measurable distance is long enough while the baseline length is halved. Moreover, the new DENSO vision sensor, shown in Fig. 4.29, is integrated with and is controlled by an ECU (URL32 2017).

4.9.2.6 Ultrasonic Sensor

The ultrasonic sensor is used in ultrasonic-distance-ranging automotive applications, such as automated parking (AP) and blind spot detection (BSD), where ultrasonic waves transmitted by the sensor are reflected by objects that are close or in the near vicinity. The system receives the reflected wave, or echo, and compares the object's echo amplitude against a threshold to detect the object. The echo for objects that are closer to the system is stronger than that for objects that are farther from the system. Hence, it is relatively common for the threshold to be varied with time.

Fig. 4.29 DENSO vision sensor

Ultrasonic sensors are installed in the front and rear bumpers and wing mirrors of a vehicle to transmit ultrasonic waves and receive the ultrasonic waves reflected back by nearby objects. An ultrasonic wave's time of flight (TOF) is used to calculate the distance to the objects to assist the driver in parking the vehicle, identifying parking spots, or detecting objects in the driver's blind spot. Up to four sensors (transducers) are installed in the front and rear bumpers, and one sensor (transducer) is installed in each wing mirror.

Ultrasonic waves generated by the sensor are a series of sinusoid pulses at carrier frequency and are characterized by sound pressure level (*SPL*), which can be expressed by:

$$\mathrm{SPL} = 20\log_{10} \cdot \left(\frac{P_{\mathrm{RMS}}}{P_{\mathrm{ref}}}\right)$$

where P_{RMS} is the *RMS*, and sound pressure P_{ref} is the reference sound pressure. The *SPL* of ultrasonic waves created by the transducer at an object depends on the object's distance from the sensor (transducer). Specifically, the pressure is inversely proportional to the distance:

$$p \sim \frac{1}{d}$$

where p is the pressure of the sound waves, and d is the distance of the object from the sensor (transducer). Tracking is a specific area of interest in the context of driver assistance systems based on ultrasound sensors such as lane change detection or blind spot surveillance systems. For given ultrasonic sensor specifications the SPL at arbitrary distance x from the sensor (transducer) can be calculated by using the distance law (URL32 2017).

4.9.3 Pros and Cons of the ADAS Sensor Suite

Today's ADASs (see Sect. 4.9.1 and Chap. 11) make use of the combination of different sensor types by combining their characteristics to achieve the most suitable performance for the respective application. Therefore, Table 4.8 shows several pros and cons of the different types of sensors used for ADAS (URL32 2017).

Table 4.8 Pros and cons of ADAS sensor types

Sensor type(s)	Pros	Cons	ADAS applications
Camera	Good lane detection	Traffic signs show very poor contrast	AP
	Usable under dark/night conditions		LDW
			OCW
			RVS
			PCCMS
			TSR
Laser	Good distance/speed detection	Poor lane detection	LDW
			LHW
	Small obstacles detectable	Poor vehicle or pedestrian detection	NFCW
			OCW
			PCCMS
LiDAR	Good obstacle detection	Poor vehicle/pedestrian detection	ACC
			BSD
	Good distance/speed detection	Smaller range than radar	PCCMS
			PPS
Radar	Accurate speed detection	Poor lane detection	ACC
	Good distance detection	Poor vehicle/pedestrian detection	BSD
			LDW
	Usable in environment with reflections	Beam blockage	PCCMS
		Big size	SaG
	Usable for short- and long-range detection		
	Waves transmitted are not affected by obstacles		
Ultrasonic	High angular range	Easily distorted by reflections on the road	ASP
	Short-range detection		BSD
		No angular position	
		No echo cancellation	
Vision	Good lane detection	Complex electronic system required for data processing	LCA
			LDW
	Good vehicle/pedestrian detection	Poor obstacle detection	LKA
			RDP
	Present images of reality Small sizes	Poor speed/distance detection	RSR
			RVS
			TSR

4.10 Trends

The automotive industry has been growing in the past decades and is an active sector with forecasts showing much more may come. Moreover, the globalization of the automotive industry has greatly accelerated during the last half of the 1990s due to the construction of important overseas facilities and mergers between giant multinational automakers. Global vehicle sales are expected to exceed 100 million units a year by 2020. There are some key trends which can be identified now:

- *Changes in Brand Loyalty*: Brand-loyal customers are rethinking their buying decisions as a result of surplus choices in the market. Impressing the customer remains harder than ever before.
- *Changes in Customer Demand*: Many customers are inclined to buy greener, fuel efficient, and sustainable vehicles. With the market launch of e-vehicles and alternative fuel, automakers became aware that the days are gone when design and style were the major decision-making factors. After Volkswagen's emission scam, customers became more cautious.
- *Changes in Mobility*: The only objective that counts is the efficient and inexpensive mobility when using a vehicle. Autonomous vehicles are not the only trend challenging the automotive industry. Views about mobility, what we can do with a vehicle, and about the status of owning a car are in transition; and the number of female buyers is increasing.
- *Resource Shortage*: According to recent growth figures, electric vehicle sales grew massively in 2016. This is roughly equivalent to the growth forecasted by Tesla Motors, where production is expected to increase from 50,000 in 2015 to 500,000 in 2020. Assuming Tesla can meet its forecasts and its current electric vehicle market share remains the same, and if each electric vehicle roughly displaces 15 barrels of oil a year, then the next oil crash will occur much later than in a pure combustion engine based scenario.
- *Technological Advances*: The global automotive industry has witnessed a lot of transformation in the last two decades with the digitization of vehicles. Linking mobile devices to the vehicle creates many options. For example, one can check how much fuel is left, the condition of the brakes, when maintenance is needed, and other features. A mobile device can also be used as a vehicle key or for applying personal settings in a rented vehicle. Thus, connecting vehicles is the next big platform for application developers. It is assumed that in 2020 approximately up to 15% of new vehicles sold could be fully autonomous, especially if one considers the activities in China in this domain by Bytons SUV business concept, in which the Byton electric car is a smart device on four wheels (URL1 2018).

The concept of connected vehicles (see Sect. 5.3), which focuses on connecting vehicles with the outside world and enhancing the onboard experience, combines telecommunication and informatics to provide various services, such as:

- Automatic parking/parking management
- Automatic toll transactions
- Live traffic updates
- Onboard entertainment
- Roadside assistance in case of accidents
- Smart routing and tracking

Therefore, the next step beyond connected vehicles will be self-driving vehicles, also called autonomous vehicles, which in the long term will revolutionize vehicle operation and the experience of driving. The most important developments on the path to self-driving cars are ADASs, which are making vehicles safer; and their gradual introduction is already improving road safety. Thus, ADAS features represent an essential evolutionary step in developing self-driving vehicles but the development and market launch of self-driving vehicles is an evolutionary step which spans a number of automotive generations. ADAS, by its very nature, perfects different essential aspects and features of automated control, one of the requirements for self-driving vehicles. It accomplishes this through independent subsystems with increasing levels of system integration, ultimately resulting in a vehicle that can drive itself.

The different kinds of ADAS are the driving force behind connected vehicles and self-driving vehicles and can be summarized by the following characteristics:

- Information and warning systems
- Function-specific automation systems
- Combined function automation systems
- Limited self-driving automation systems
- Full self-driving automation systems

Security is becoming a major concern as vehicles are beginning to communicate with each other and with the road infrastructure installations as well as with traffic signs and traffic lights. Thus, connected and self-driving vehicles will need protection from malicious intrusion, i.e., vehicle hacking, which is described in detail in Chap. 6. In this regard the design and manufacturing of vehicular components and systems as well as vehicles itself require to follow a new design and manufacturing paradigm, which can be stated as security by design, as it was introduced by the German Industry 4.0 Platform (URL2 2018).

4.11 Exercises

What is meant by the term *mechatronics*?
Describe the characteristics of mechatronic systems.
What is meant by the term *intelligent control*?
Describe the characteristics of intelligent control.
What is meant by the term *cyber-physical systems*?

Describe the characteristics of cyber-physical systems.
What is meant by the term *automotive electronics*?
Describe the characteristics of automotive electronics.
What is meant by the term *body electronics*?
Describe the characteristics of body electronics.
What is meant by the term *chassis electronics*?
Describe the characteristics of chassis electronics.
What is meant by the term *comfort electronics*?
Describe the characteristics of comfort electronics.
What is meant by the term *driver assistance electronics*?
Describe the characteristics of driver assistance electronics.
What is meant by the term *electronic control unit*?
Describe the characteristics of an electronic control unit.
What is meant by the term *entertainment/infotainment electronics*?
Describe the characteristics of entertainment/infotainment electronics.
What is meant by the term *passive safety electronics*?
Describe the characteristics of passive safety electronics.
What is meant by the term *bus system*?
Describe the characteristics of bus systems.
What is meant by the term *entertainment electronics?*
Describe the characteristics of entertainment electronics.
What is meant by the term *infotainment electronics?*
Describe the characteristics of infotainment electronics.
What is meant by the term *sensor technology*?
Describe the characteristics of sensor technologies.
What is meant by the term *signal-to-noise ratio*?
Describe the signal-to-noise ratio mathematically and explain it.
What is meant by the term *active sensor*?
Describe the characteristics of an active sensor.
What is meant by the term *microelectromechanical systems*?
Describe the characteristics of microelectromechanical systems.
What is meant by the term *sensor node*?
Describe the characteristics of sensor nodes.
What is meant by the term *sensor data fusion*?
Describe the characteristics of sensor data fusion.
What is meant by the term *sensor network*?
Describe the characteristics of sensor networks.
What is meant by the term *analog-to-digital conversion*?
Describe the analog-to-digital conversion process.
What is meant by the term *ADC resolution characteristic*?
Describe the characteristics of ADC resolution qualitatively.
What is meant by the term *bus system*?
Describe the characteristics of bus systems.

What is meant by the term *CAN bus system*?
Describe the characteristics of the CAN bus systems.
What is meant by the term *LIN bus system*?
Describe the characteristics of the LIN bus systems.
What is meant by the term *FlexRay bus system*?
Describe the characteristics of the FlexRay bus systems.
What is meant by the term *functional safety*?
Describe the characteristics of functional safety.
What is meant by the term *safe failure fraction*?
Describe the characteristics of the safe failure fraction.
What is meant by the term *failure mode and effects and criticality analysis (FMECA)*?
Describe the characteristics of the failure mode and effects and criticality analysis.
What is meant by the term *agile software development*?
Describe the characteristics of the agile software development.
What is meant by the term *automotive spice*?
Describe the characteristics of the automotive spice.
What is meant by the term *ASAM*?
Describe the characteristics of ASAM.
What is meant by the term *model-based development*?
Describe the characteristics of model-based development.
What is meant by the term *rapid prototyping*?
Describe the characteristics of rapid prototyping.
What is meant by the term *model-in-the-loop test*?
Describe the characteristics of the model-in-the-loop test.
What is meant by the term *hardware-in-the-loop test*?
Describe the characteristics of the hardware-in-the-loop test.
What is meant by the term *AUTOSAR*?
Describe the characteristics of the AUTOSAR.
What is meant by the term *adaptive AUTOSAR platform*?
Describe the characteristics of the adaptive AUTOSAR platform.
What is meant by the term *GENIVI*?
Describe the characteristics of GENIVI.
What is meant by the term *advanced driver assistance system*?
Describe the characteristics of advanced driver assistance systems.
What is meant by the term *advanced driver assistance system (ADAS)*?
Describe the characteristics of advanced driver assistance systems functionalities.
What is meant by the term *advanced driver assistance system sensors*?
Describe the characteristics of advanced driver assistance systems sensors.
What is meant by the term *original equipment manufacturer (OEM)*?
Describe the characteristics of OEM.
What is meant by the term *Tier 1 supplier*?
Describe the characteristics of Tier 1 supplier.
What is meant by the term *connected car?*
Describe the characteristics of connected cars.

What is meant by the term *autonomous vehicle*?
Describe the characteristics of autonomous vehicles.
What is meant by the term *connected car gateway*?
Describe the characteristics of the connected car gateway.

References and Further Reading

(AA1Car 2016) Troubleshoot Automatic Climate Control Systems. 2016 http://www.aa1car.com/library/automatic_climate_control.htm

(Barr 2004) Barr, D.: Supervisory Control and Data Acquisition (SCADA) Systems. NCS Technical Information Bulletin 04-1, 2004

(Bechmann et al. 2015) Bechmann, R., Scherk, M., Heimann, R., Schäfer, R.: Trend Analysis: Connected Car 2015. MBtech Consulting GmbH, 2015. Available from: https://www.yumpu.com/en/document/view/10955202/trend-analysis-connected-car-2015-mbtech-group

(Berlin and Gabriel 1997) Berlin, A. A., Gabriel, K. J.: Distributed MEMS: New Challenges for Computation. In: IEEE Computational Science and Engineering, Vol. 4, No. 1, pp. 12–16, 1997

(Biddlecombe 2005) Biddlecombe, E.: BBC News 17.11.2005

(Chaouchi 2010) Chaouchi, H.: The Internet of Things – Connecting Objects to the Web. J. Wiley Publ., 2010

(DADSS 2016) Inventing a world without drunk driving; www.dadss.org

(Gironimo et al. 2009) Gironimo, D., Lopez, A. M., Sappa, A. D., Graf, T.: Survey of Pedestrian Detection for Advanced Driver Assist Systems. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, Issue 7, pp. 1239–1258, 2009

(Golatowski et al. 2003) Golatowski, F., Blumenthal, J., Handy, M., Haase, M., Burchardt, H., Timmermann, D.: Service-Oriented Software Architecture for Sensor Networks. In: Proc. of Internat. Workshop on Mobile Computing (IMC), pp. 93–98, 2003

(Grosan and Abraham 2007) Grosan, C., Abraham, A.: Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews. In: Hybrid Evolutionary Algorithms, Studies in Computational Intelligence, Springer Publ. 2007

(Heinecke et al. 2003) Heinecke, H., Schnelle, K.-P-, Fennel. H., Bortolazzi, J., Lundh, L., Leflour, J., Mate, J.-L., Nishikawa, K., Scharnhorst, T.: AUTomotive Open System ARchitecture – An Industry-Wide Initiative to Manage the Complexity of Emerging Automotive E/E-Architectures. Convergence Transportation Electronics Association, 2004

(Holtz and Möller 2017) Holtz, S., Möller, D. P. F.: Agile Software Development Use Case Example: Interface Renewal for Automated Loading of Tank Trucks. In: Proceed. IEEE/EIT Conference 2017, pp.115–120

(Mercer Management Consulting 2001) Mercer Management Consulting and Hypovereinsbank. Study - Automotive Technology 2010 (in German), Munich, 2001

(Möller 2003) Möller D. P. F.: Computer Structures – Fundamentals of Computer Science (in German), Springer Publ. 2003

(Möller 2016) Möller, D. P. F.: Guide to Computing Fundamentals in Cyber-Physical Systems – Concepts, Design Methods, and Applications. Springer Publ., 2016

(Moonen et al. 2005) Moonen, A., von den Berg, R., Bekoooij, M., Bhullar, H., van Meerbergen, J.: A Multi-Core Architecture for In-Car Digital Entertainment. http://www.es.ele.tue.nl/epicurus/publications/gspx05.pdf

(Moritz et al. 2011) Moritz, R., Ulrich, T., Thiele, L.: Evolutionary Exploration of E/E-Architectures in Automotive Design, pp. 361–365. In: Operations Research Proceedings, 2011, Eds.: Klatte, D., Lüth, H.-J., Schmedders, K., Springer Publ., 2011

(Ning 2013) Ning, H.: Unit and Ubiquitous Internet of Things, CRC Press, 2013

(Pala and Inanc 2007) Pala, Z., Inanc, N.: Smart Parking Applications Using RFID Technology. In: RFID Eurasia. 1st Annual Conference, 2007 DOI: https://doi.org/10.1109/RFIDEURASIA.2007.4368108

(Poslad 2009) Poslad, S.: Ubiquitous Computing – Smart Devices, Environments and Interactions. John Wiley and Sons Publ., 2009

(Rosengren 1995) Rosengren, L. G.: Driver assistance and co-operative driving. In: Proceedings of the First World Congress on Advanced Transport Telematics and Intelligent Vehicle Highway Systems, pp. 1613-1622, Artech House Publ., 1995

(Schäuffele and Zurawka 2016) Schäuffele, J., Zurawka, T.: Automotive Software Engineering: Efficient Use of Basics, Processes, Methods and Tools (in German), Springer Publ. 2016

(ST Microelectronics 2013) Complete car door module – AN 2334 Application Note. ST Microelectronics, 2013

(Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from: https://www.sasken.com/sites/default/files/files/white_paper/Sasken-Whitepaper-Connected%20Cars%20Challenges.pdf

(Vemuri 2012) Vemuri, A. T.: Using a fixed threshold in ultrasonic distance-ranging automotive applications. In: Analog Applications Journal, Vol 30, pp. 19-23. 2012

(Yu and Gen 2010) Yu, X., Gen, M.: Introduction to Evolutionary Algorithms. Springer Publ. 2010

(Zhang and Mi 2011) Zhang, X., Mi, C.: Vehicle Power Management – Modeling, Control and Optimization. Springer Publ. 2011

(Zhao and Guibas 2004) Zhao F., Guibas, L.: Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann Publ. 2004

Links

(URL1 2013) https://www.nxp.com/docs/en/white-paper/BODYDELECTRWP.pdf
(URL1 2017) https://en.wikipedia.org/wiki/Body_control_module
(URL2 2017) https://en.wikipedia.org/wiki/Remote_keyless_system
(URL3 2017) http://www.meeknet.co.uk/e38/E38_Sunroof_Description.pdf
(URL4 2017) https://en.wikipedia.org/wiki/Anti-lock_braking_system
(URL5 2017) https://en.wikipedia.org/wiki/Electronic_brakeforce_distribution
(URL6 2017) https://en.wikipedia.org/wiki/Electronic_stability_control
(URL7 2017) https://en.wikipedia.org/wiki/Airbag
(URL8 2017) https://en.wikipedia.org/wiki/Hill_Descent_Control_system
(URL9 2017) https://en.wikipedia.org/wiki/Advanced_driver_assistance_systems
(URL10 2017) https://en.wikipedia.org/wiki/Power_steering
(URL11 2017) https://www.mems-exchange.org/MEMS/what-is.html
(URL12 2017) http://en.wikipedia.org/wiki/Sensor_node
(URL13 2017) https://community.nxp.com/docs/DOC-335306
(URL14 2017) https://en.wikipedia.org/wiki/Safety_integrity_level
(URL15 2017) https://www.powerspex.nl/en/wat-is/sil/
(URL16 2017) https://webstore.iec.ch/publication/5517
(URL17 2017) https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis
(URL18 2017) https://wiki.asam.net/display/STANDARDS/ASAM+MCD-2+D
(URL19 2017) http://www.ni.com/white-paper/10343/en/
(URL20 2017) www.dSPACE.org
(URL21 2017) www.autosar.org
(URL22 2017) https://www.autosar.org/standards/adaptive-platform/
(URL23 2017) en.m.wikipedia.org/wiki/Advanced_driver_assistance_systems
(URL24 2017) https://in.mathworks.com/solutions/automotive/advanced-driver-assistance-systems.html

(URL25 2017) www.add2.co.uk/application/model-in-the-loop-testing-application
(URL26 2017) https://trimis.ec.europa.eu/project/advanced-driver-assistance-systems-europe-thematic-network-cluster-project
(URL27 2017) http://www.continentalautomotive.com/www/automotive_de_en/themes/passenger_cars/chassis_safety/adas/ldw_en.html
(URL28 2017) http://www.continentalautomotive.com/www/automotive_de_en/themes/passenger_cars/chassis_safety/adas/rdp_en.html
(URL29 2017) http://leddartech.com/app/uploads/dlm_uploads/2016/02/Solution-Overview-Leddar-for-automotive-ADAS.pdf
(URL30 2017) http://www.sabertek.com/automotive-radar.html
(URL31 2017) http://www.ti.com/lsds/ti/applications/automotive/adas/overview.page
(URL32 2017) https://www.utwente.nl/en/et/aida/education/Rapport_MP.pdf
(URL33 2017) https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/facing-digital-disruption-in-mobility-as-a-traditional-auto-player
(URL1 2018) https://t3n.de/news/byton-chef-breitfeld-zukunft-autoindustrie-1062843/ (in German)
(URL2 2018) https://www.plattform-i40.de/I40/Navigation/DE/Industrie40/Handlungsfelder/Sicherheit/sicherheit.html